ACL 2007

Proceedings of the Workshop on BioNLP 2007

Biological, Translational, and Clinical Language Processing

June 29, 2007
Prague, Czech Republic

The Association for Computational Linguistics

Production and Manufacturing by
Omnipress
2600 Anderson Street
Madison, WI 53704
USA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Biological, translational, and clinical language processing

K. BRETONNEL COHEN, DINA DEMNER-FUSHMAN, CAROL FRIEDMAN, LYNETTE HIRSCHMAN, AND JOHN P. PESTIAN

1 Background and goals of the workshop

Natural language processing has a long history in the medical domain, with research in the field dating back to at least the early 1960s. In the late 1990s, a separate thread of research involving natural language processing in the genomic domain began to gather steam. It has become a major focus of research in the bioinformatics, computational biology, and computational linguistics communities. A number of successful workshops and conference sessions have resulted, with significant progress in the areas of named entity recognition for a wide range of key biomedical classes, concept normalization, and system evaluation. A variety of publicly available resources have contributed to this progress, as well.

Recently, the widely recognized disconnect between basic biological research and patient care delivery stimulated development of a new branch of biomedical research—translational medicine. Translational medicine, sometimes defined as the facilitation of "bench-to-bedside" transmission of knowledge, has become a hot topic, with a National Center for Biocomputing devoted to this theme established last year.

This workshop has the goal of addressing and bringing together these three threads in biomedical natural language processing, or "BioNLP:" biological, translational, and clinical language processing.

2 Submissions and acceptance rate

The workshop received 59 submissions—almost twice the number of submissions of any previous BioNLP workshop or conference session that we are aware of (31 for last year's PSB session on *New frontiers in text mining*, [18]). The submissions covered a wide range of topics from most areas of natural language processing and from both the clinical and the genomics domains. There were 48 full-paper submissions and 11 poster submissions. A strong program committee comprising members of the BioNLP community from North America, Europe, and Asia provided three reviews for each submission. Out of the many strong pieces of work submitted, fourteen papers were accepted for oral presentation, as well as nineteen posters. The subjects of the papers fell into five or six broad categories:

- Syntax
- Lexical semantics and terminology

- Named entity recognition and word sense disambiguation
- Information extraction
- Usability and user interface design
- Shared tasks

3 Themes in the papers

A number of trends were notable in the accepted papers. Compared to past years, the number of papers on gene mention recognition was quite small. We did see strong work on named entity recognition for new semantic classes, as well as on the gene normalization task.

There were also a number of papers on syntactic topics. Other than the pioneering work of the GENIA group some years ago and two recent papers on parser evaluation [4, 5], there has been little work on syntax in biomedical NLP to date. However, three papers on syntactic topics appear in this proceedings volume–[12, 14, 15]. [15] is especially unique in dealing with an actual clinical application.

Lexical semantics and terminology also figured heavily in this year's workshop. [16] discussed the gene symbol disambiguation problem. [8] presented a system for mapping clinical terminology to lay terminology. [6] presented work on the development of a corpus annotated with a semantic class of entity that has previously received scant attention in the field. [7] explored the potential of domain-specific semantic roles for use in information extraction and document classification. It is notable that there were no papers on the classic "gene mention" problem; although it is clear that gene mention recognition is not yet a solved problem [17], it is encouraging that work in this area is progressing, and our sole paper on this task dealt with the more complex problem of recognizing nested entities [1].

The work on information extraction that appeared this year was often quite innovative. Chapman described an extension of the NegEx algorithm to extract various kinds of context-establishing information. [11] presented work on an unsupervised method for protein-protein interaction detection, using graph-based mutual reinforcement.

Finally, three papers demonstrated the continued contribution of shared tasks to progress in the field. [13] described a shared task that resulted in the public availability of a large document collection of clinical texts. [2] used the data from that task and the associated evaluation itself to test a number of hypotheses regarding the differences between published and clinical texts and regarding the portability of text mining systems to new domains. [16] (also mentioned above in the context of lexical semantics and terminology) utilitized data from the BioCreative shared tasks as a source of test data.

There were an encouraging number of papers that focussed on the usability and accessibility of text mining and of information access systems. [9] describes a novel search interface, and provides valuable insight into the design of usability studies. [8] (like [16], also mentioned above in the context of lexical semantics and terminology) described a system that aids in the process of making medical information more intelligible to the lay public.

There was a notable broadening of the types of genres of textual inputs that this year's papers dealt with. In previous years, most work has tended to deal with abstracts drawn from PubMed/MEDLINE or with ontologies, with occasional forays into longer texts, such as full-text journal articles, or shorter ones,

such as GeneRIFs. This year's workshop contains work on newsfeeds [7], clinical data [2, 3, 12, 13], full text [9], and speech [15]—a genre heretofore essentially entirely neglected in the BioNLP field.

Finally, the accepted posters reflect an enormously fertile field. The poster session includes much work that would have had oral presentations in a less-competitive meeting. The topics of the posters cover a range of subjects every bit as diverse and interesting as the work with oral presentation; the executive committee regrets that time constraints did not allow for more of it to have oral presentations.

4 Acknowledgements

The biggest debt owed by the organizers of a workshop like this is to the authors who graciously chose BioNLP 2007 as the venue in which to share the fruits of the countless hours of research that went into the work submitted for consideration. The next-biggest debt is, without question, to the many program committee members (listed elsewhere in this volume); they produced almost 180 reviews, on a tight review schedule and with an admirable level of insight. We also thank Simone Teufel, the ACL Workshop Chair, and Su Jian, the Publications Chair, for their patient responses to many inquiries over the past few months. Finally, Laura Grushcow provided hours of invaluable assistance in the preparation of the Proceedings volume.

References

[1] Alex, Beatrice; Barry Haddow; and Claire Grover (2007) Recognising nested named entities in biomedical text. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 65–72.

[2] Aronson, Alan R.; Olivier Bodenreider; Dina Demner-Fushman; Kin Wah Fung; Vivian K. Lee; James G. Mork; Aurélie Névéol; Lee Peters; and Willie J. Rogers (2007) From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches (2007) *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 105–112.

[3] Chapman, Wendy; David Chu; and John N. Dowling (2007) ConText: An algorithm for identifying contextual features from clinical text. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 81–88.

[4] Clegg, Andrew B.; and Adrian J. Shepherd (2005) Evaluating and integrating treebank parsers on a biomedical corpus. *Proceedings of the Association for Computational Linguistics workshop on software 2005.*

[5] Clegg, Andrew B.; and Adrian J. Shepherd (2007) Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics* 8(24).

[6] Corbett, Peter; Colin Batchelor; and Simone Teufel (2007) Annotation of chemical named entities. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 57–64.

[7] Doan, Son; Ai Kawazoe; and Nigel Collier (2007) The role of roles in classifying annotated biomedical text. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 17–24.

[8] Elhadad, Noëmie; and Komal Sutaria (2007) Mining a lexicon of technical terms and lay equivalents. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 49–56.

[9] Hearst, Marti A.; Anna Divoli; Jerry Ye; and Michael A. Wooldridge (2007) Exploring the efficacy of caption search for bioscience journal search interfaces. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 73–80.

[10] Liu, Haibin; Christian Blouin; and Vlado Keselj (2007). An unsupervised method for extracting domain-specific affixes in biological literature. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 33–40.

[11] Madkour, Amgad; Kareem Darwish; Hany Hassan; Ahmed Hassan; and Ossama Emam (2007) BioNoculars: extracting protein-protein interactions from biomedical text. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 89–96.

[12] McInnes, Bridget T.; Ted Pedersen; and Serguei V. Pakhomov (2007) Determining the syntactic structure of medical terms in clinical notes. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 9–16.

[13] Pestian, John P.; Christopher Brew; Pawel Matykiewicz; DJ Hovermale; Neil Johnson; K. Bretonnel Cohen; and Wlodiszlaw Duch (2007) A shared task involving multi-label classification of clinical free text. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 97–104.

[14] Pyysalo, Sampo; Filip Ginter; Katri Haverinen; Veronika Laippala; Juho Heimonen; and Tapio Salakoski (2007) On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 25–32.

[15] Roark, Brian; Margaret Mitchell; and Kristy Hollingshead (2007) Syntactic complexity measures for detecting Mild Cognitive Impairment. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. 1–9.

[16] Xu, Hua; Jung-Wei Fan; and Carol Friedman (2007) Combining multiple evidence for gene symbol disambiguation. *BioNLP 2007: Biological, translational, and clinical language processing,* pp. xx–yy.

[17] Wilbur, W. John; Lawrence Smith; and Lorraine Tanabe (2007) BioCreative 2: Gene mention task. In Lynette Hirschman, Martin Krallinger, and Alfonso Valencia, eds.: *Proceedings of the second BioCreative challenge evaluation workshop,* pp. 7–16.

[18] Zweigenbaum, Pierre; Dina Demner-Fushman; Hong Yu; and K. Bretonnel Cohen (2007) New frontiers in biomedical text mining. *Pacific Symposium on Biocomputing* 12:205-208.

Organizers

Chairs:

K. Bretonnel Cohen, University of Colorado School of Medicine
Dina Demner-Fushman, Lister Hill National Center for Biomedical Communications
Carol Friedman, Columbia Universtity
Lynette Hirschman, MITRE
John Pestian, Computational Medicine Center, University of Cincinnati, Cincinnati Children's Hospital Medical Center

Program Committee:

Sophia Ananiadou, NaCTeM
Lan Aronson, NLM
Breck Baldwin, alias-i
Sabine Bergler, Concordia University
Catherine Blake, U. North Carolina Chapel Hill Christian Blaschke, bioalma
Olivier Bodenreider, NLM
Chris Brew, Ohio State University
Allen Browne, NIH
Bob Carpenter, alias-i
Jeffrey Chang, Duke
Wendy Chapman, University of Pittsburgh
Aaron Cohen, Oregon Health and Science University
Nigel Collier, National Institute of Informatics
Anna Divoli, UC Berkeley
Noemie Elhadad, CCNY
Kristofer Franzen, SICS
Udo Hahn, JULIE Lab, Jena University
Peter Haug, University of Utah
Marti Hearst, UC Berkeley
George Hripcsak, Columbia University
John Hurdle, U. Utah
Steve Johnson, Columbia University
Michael Krauthammer, Yale
Marc Light, Thomson Corporation
Alex Morgan, Stanford
Serguei Pakhomov, Mayo Clinic
Martha Palmer, University of Colorado at Boulder
Dietrich Rebholz-Schuhmann, EBI
Tom Rindflesch, NLM
Patrick Ruch, U. and Hospitals of Geneva
Jasmin Saric, Boehringer Ingelheim
Guergana Savova, Mayo Clinic

Hagit Shatkay, Queens University
Larry Smith, NLM
Padmini Srinivasan, U. Iowa
Lorrie Tanabe, NLM
Jun'ichi Tsujii, University of Tokyo and NaCTeM
Alfonso Valencia, CNIO
Karin Verspoor, Los Alamos National Laboratory
Bonnie Webber, University of Edinburgh
Pete White, Children's Hospital of Philadelphia
W. John Wilbur, NLM
Limsoon Wong, National U. of Singapore
Hong Yu, University of Wisconsin
Pierre Zweigenbaum, LIMSI

Additional Reviewers:

Guy Divita, NLM
Jung-wei Fan, Columbia University
Helen L. Johnson, University of Colorado School of Medicine
Sriharsha Veeramachaneni, Thomson Corporation
HaThuc Viet, University of Iowa
Hua Xu, Columbia University

Invited Speaker:

Alfonso Valencia, CNIO

Table of Contents

POSTERS

Conference Program

Friday, June 29, 2007

Welcome and opening remarks

8:30–8:40 BioNLP 2007: Biological, translational, and clinical language processing

Syntax in BioNLP

8:40–9:00 *Syntactic complexity measures for detecting Mild Cognitive Impairment*
Brian Roark, Margaret Mitchell and Kristy Hollingshead

9:00–9:20 *Determining the Syntactic Structure of Medical Terms in Clinical Notes*
Bridget McInnes, Ted Pedersen and Serguei Pakhomov

9:20–9:40 *The Role of Roles in Classifying Annotated Biomedical Text*
Son Doan, Ai Kawazoe and Nigel Collier

9:40–10:00 *On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA*
Sampo Pyysalo, Filip Ginter, Veronika Laippala, Katri Haverinen, Juho Heimonen and Tapio Salakoski

Terminology and computational lexical semantics in BioNLP, Part I

10:00–10:20 *An Unsupervised Method for Extracting Domain-specific Affixes in Biological Literature*
Haibin Liu, Christian Blouin and Vlado Keselj

10:20–10:40 *Combining multiple evidence for gene symbol disambiguation*
Hua Xu, Jung-Wei Fan and Carol Friedman

10:45–11:15 COFFEE BREAK

Friday, June 29, 2007 (continued)

Terminology and computational lexical semantics in BioNLP, Part II

11:15–11:35 *Mining a Lexicon of Technical Terms and Lay Equivalents*
Noemie Elhadad and Komal Sutaria

11:35–11:55 *Annotation of Chemical Named Entities*
Peter Corbett, Colin Batchelor and Simone Teufel

11:55–12:15 *Recognising Nested Named Entities in Biomedical Text*
Beatrice Alex, Barry Haddow and Claire Grover

12:30–2:30 LUNCH

Keynote speech

2:30–3:25 Alfonso Valencia

Interfaces and usability in BioNLP

3:25–3:45 *Exploring the Efficacy of Caption Search for Bioscience Journal Search Interfaces*
Marti Hearst, Anna Divoli, Ye Jerry and Michael Wooldridge

3:45–4:15 COFFEE BREAK

Information extraction in BioNLP

4:15–4:35 *ConText: An Algorithm for Identifying Contextual Features from Clinical Text*
Wendy Chapman, John Dowling and David Chu

4:35–4:55 *BioNoculars: Extracting Protein-Protein Interactions from Biomedical Text*
Amgad Madkour, Kareem Darwish, Hany Hassan, Ahmed Hassan and Ossama Emam

Friday, June 29, 2007 (continued)

Shared tasks in BioNLP

4:55–5:15 *A shared task involving multi-label classification of clinical free text*
John P. Pestian, Chris Brew, Pawel Matykiewicz, DJ Hovermale, Neil Johnson, K. Bretonnel Cohen and Wlodzislaw Duch

5:15–5:35 *From indexing the biomedical literature to coding clinical text: experience with MTI and machine learning approaches*
Alan R. Aronson, Olivier Bodenreider, Dina Demner-Fushman, Kin Wah Fung, Vivian K. Lee, James G. Mork, Aurelie Neveol, Lee Peters and Willie J. Rogers

Poster session

5:35–7:00 Poster session

Automatically Restructuring Practice Guidelines using the GEM DTD
Amanda Bouffier and Thierry Poibeau

A Study of Structured Clinical Abstracts and the Semantic Classification of Sentences
Grace Chung and Enrico Coiera

Automatic Code Assignment to Medical Text
Koby Crammer, Mark Dredze, Kuzman Ganchev, Partha Pratim Talukdar and Steven Carroll

Interpreting comparative constructions in biomedical text
Marcelo Fiszman, Dina Demner-Fushman, Francois M. Lang, Philip Goetz and Thomas C. Rindflesch

The Extraction of Enriched Protein-Protein Interactions from Biomedical Text
Barry Haddow and Michael Matthews

What's in a gene name? Automated refinement of gene name dictionaries
Jörg Hakenberg

Exploring the Use of NLP in the Disclosure of Electronic Patient Records
David Hardcastle and Catalina Hallett

BaseNPs that contain gene names: domain specificity and genericity
Ian Lewin

Friday, June 29, 2007 (continued)

Challenges for extracting biomedical knowledge from full text
Tara McIntosh and James R. Curran

Adaptation of POS Tagging for Multiple BioMedical Domains
John E. Miller, Manabu Torii and K. Vijay-Shanker

Information Extraction from Patients' Free Form Documentation
Agnieszka Mykowiecka and Malgorzata Marciniak

Automatic Indexing of Specialized Documents: Using Generic vs. Domain-Specific Document Representations
Aurelie Neveol, James G. Mork and Alan R. Aronson

Developing Feature Types for Classifying Clinical Notes
Jon Patrick, Yitao Zhang and Yefeng Wang

Quantitative Data on Referring Expressions in Biomedical Abstracts
Michael Poprat and Udo Hahn

Discovering contradicting protein-protein interactions in text
Olivia Sanchez and Massimo Poesio

Marking time in developmental biology
Gail Sinclair and Bonnie Webber

Evaluating and combining and biomedical named entity recognition systems
Andreas Vlachos

Unsupervised Learning of the Morpho-Semantic Relationship in MEDLINE
W. John Wilbur

Reranking for Biomedical Named-Entity Recognition
Kazuhiro Yoshida and Jun'ichi Tsujii

Syntactic complexity measures for detecting Mild Cognitive Impairment

Brian Roark, Margaret Mitchell and **Kristy Hollingshead**
Center for Spoken Language Understanding
OGI School of Science & Engineering
Oregon Health & Science University
Beaverton, Oregon, 97006 USA
{roark,meg.mitchell,hollingk}@cslu.ogi.edu

Abstract

We consider the diagnostic utility of various syntactic complexity measures when extracted from spoken language samples of healthy and cognitively impaired subjects. We examine measures calculated from manually built parse trees, as well as the same measures calculated from automatic parses. We show statistically significant differences between clinical subject groups for a number of syntactic complexity measures, and these differences are preserved with automatic parsing. Different measures show different patterns for our data set, indicating that using multiple, complementary measures is important for such an application.

1 Introduction

Natural language processing (NLP) techniques are often applied to electronic health records and other clinical datasets. Another potential clinical use of NLP is for processing patient language samples, which can be used to assess language development (Sagae et al., 2005) or the impact of neurodegenerative impairments on speech and language (Roark et al., 2007). In this paper, we present methods for automatically measuring syntactic complexity of spoken language samples elicited during neuropsychological exams of elderly subjects, and examine the utility of these measures for discriminating between clinically defined groups.

Mild Cognitive Impairment (MCI), and in particular amnestic MCI, the earliest clinically defined stage of Alzheimer's-related dementia, often goes undiagnosed due to the inadequacy of common screening tests such as the Mini-Mental State Examination (MMSE) for reliably detecting relatively subtle impairments. Linguistic memory tests, such as word list and narrative recall, are more effective than the MMSE in detecting MCI, yet are still individually insufficient for adequate discrimination between healthy and impaired subjects. Because of this, a battery of examinations is typically used to improve psychometric classification. Yet the summary recall scores derived from these linguistic memory tests (total correctly recalled) ignore potentially useful information in the characteristics of the spoken language itself.

Narrative retellings provide a natural, conversational speech sample that can be analyzed for many of the characteristics of speech and language that have been shown to discriminate between healthy and impaired subjects, including syntactic complexity (Kemper et al., 1993; Lyons et al., 1994) and mean pause duration (Singh et al., 2001). These measures go beyond simply measuring fidelity to the narrative, thus providing key additional dimensions for improved diagnosis of impairment. Recent work (Roark et al., 2007) has shown significant differences between healthy and MCI groups for both pause related and syntactic complexity measures derived from transcripts and audio of narrative recall tests. In this paper, we look more closely at syntactic complexity measures.

There are two key considerations when choosing how to measure syntactic complexity of spoken language samples for the purpose of psychometric evaluation. First and most importantly, the syntactic complexity measures will be used for discrimination between groups, hence high discriminative utility is desired. It has been demonstrated in past studies (Cheung and Kemper, 1992) that many competing measures are in fact very highly correlated, so it may be the case that many measures are equally discriminative. For this reason, previous results (Roark et al., 2007) have focused on a single syntactic complexity metric, that of Yngve (1960).

A second key consideration, however, is the fidelity of the measure when derived from transcripts via automatic parsing. Different syntactic complexity measures rely on varying levels of detail from

BioNLP 2007: Biological, translational, and clinical language processing, pages 1–8,
Prague, June 2007.

the parse tree. Some syntactic complexity measures, such as that of Yngve (1960), make use of unlabeled tree structures to derive their scores; others, such as that of Frazier (1985), rely on labels within the tree, in addition to the tree structure, to provide the scores. Given these different uses of detail, some measures may be less reliable with automation, hence dis-preferred in the context of automated evaluation. Ideally, simple, easy-to-automate measures with high discriminative utility are preferred.

In the current paper, we demonstrate that various syntactic complexity measures capture complementary systematic differences between subject groups, suggesting that the best approach to discriminating between healthy and impaired subjects is to collect various measures, as a way of capturing language "signatures" of the impairment.

For many measures of syntactic complexity, the nature of the syntactic annotation is critical – different conventions of structural annotation will yield different scores. We will thus spend the next section briefly detailing the syntactic annotation conventions that were followed for this work. This is followed by a section describing a range of complexity measures to be derived from these annotations. Finally, we present empirical results on the samples of spoken narrative retellings.

2 Syntactic annotation

For manual syntactic annotation of collected data (see Section 4), we followed the syntactic annotation conventions of the well-known Penn Treebank (Marcus et al., 1993). This provides several key benefits. First, there is an extensive annotation guide that has been developed, not just for written but also for spoken language, so that consistent annotation was facilitated. Second, the large out-of-domain corpora, in particular the 1 million words of syntactically annotated Switchboard telephone conversations, provide a good starting point for training domain adapted parsing models. Finally, we can use multiple domains for evaluating the correlations between various syntactic complexity measures.

There are characteristics of Penn Treebank annotation that can impact syntactic complexity scoring. First, prenominal modifiers are typically grouped in a flat constituent with no internal structure. This annotation choice can result in very long noun phrases (NPs) which pose very little difficulty in terms of human processing performance, but can inflate complexity measures that measure deviation from right-branching structures, such as that of Yngve (1960). Second, in spoken language annotations, a *reparandum*[1] is denoted with a special non-terminal category EDITED. For this paper, we remove from the tree these non-terminals, and the structures underneath them, prior to evaluating syntactic complexity.

3 Syntactic complexity

There is no single agreed-upon measurement of syntactic complexity. A range of measures have been proposed, with different primary considerations driving the notion of complexity for each. Many measures focus on the order in which various constructions are acquired by children learning the syntax of their native language – later acquisitions being taken as higher complexity. Examples of this sort of complexity measure are: mean length of utterance (MLU), which is typically measured in morphemes (Miller and Chapman, 1981); the Index of Productive Syntax (Scarborough, 1990), a multi-point scale which has recently been automated for child-language transcript analysis (Sagae et al., 2005); and Developmental Level (Rosenberg and Abbeduto, 1987), a 7-point scale of complexity based on the presence of specific grammatical constructions. Other approaches have relied upon the right-branching nature of English syntactic trees (Yngve, 1960; Frazier, 1985), under the assumption that deviations from that correspond to more complexity in the language. Finally, there are approaches focused on the memory demands imposed by "distance" between dependent words (Lin, 1996; Gibson, 1998).

3.1 Yngve scoring

The scoring approach taken in Yngve (1960) is related to the size of a "first in/last out" stack at each word in a top-down, left-to-right parse derivation. Consider the tree in Figure 1. If we knew exactly which productions to use, the parse would begin with an S category on the stack and advance as follows: pop the S and push VP and NP onto the stack; pop NP and push PRP onto the stack; pop PRP from the stack; pop VP from the stack and push NP and VBD onto the stack; and so on. At the point when the word 'she' is encountered, only VP remains on the stack of the parser. When 'was'

[1] A reparandum is a sequence of words that are aborted by the speaker, then *repaired* within the same utterance.

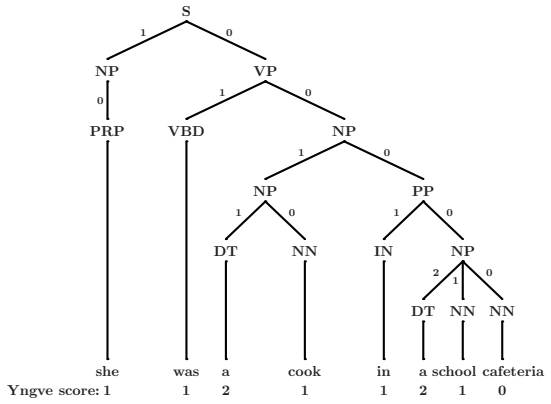


Figure 1: Parse tree with branch scores for Yngve scoring.

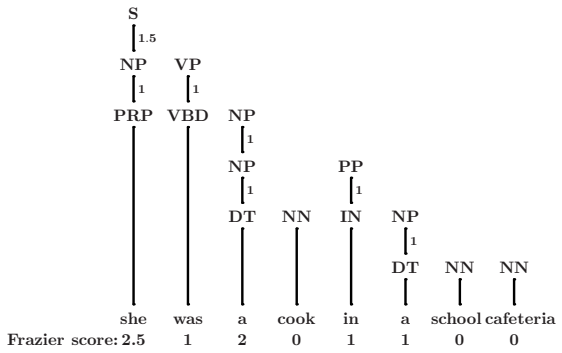


Figure 2: Parse tree fragments with scores for Frazier scoring.

is reached, just NP is on the stack. Thus, the Yngve score for these two words is 1. When the next word 'a' is reached, however, there are two categories on the stack: PP and NN, so this word receives an Yngve score of 2. Stack size has been related by some (Resnik, 1992) to working memory demands, although it most directly measures deviation from right-branching trees.

To calculate the size of the stack at each word, we can use the following simple algorithm. At each node in the tree, label the branches from that node to each of its children, beginning with zero at the rightmost child and continuing to the leftmost child, incrementing the score by one for each child. Hence, each rightmost branch in the tree of Figure 1 is labeled with 0, the leftmost branch in all binary nodes is labeled with 1, and the leftmost branch in the ternary node is labeled with 2. Then the score for each word is the sum of the branch scores from the root of the tree to the word.

Given the score for each word, we can then derive an overall complexity score by summing them or taking the maximum or mean. For this paper, we report mean scores for this and other word-based measures, since we have found these means to provide better performing scores than either total sum or maximum. For the tree in Figure 1, the maximum is 2, the total is 9 and the mean over 8 words is $1\frac{1}{8}$.

3.2 Frazier scoring

Frazier (1985) proposed an approach to scoring syntactic complexity that traces a path from a word up the tree until reaching either the root of the tree or the lowest node which is not the leftmost child of its parent.[2] For example, Figure 2 shows the tree from Figure 1 broken into distinct paths for each word in the string. The first word has a path up to the root, while the second word just up to the VP, since the VP has an NP sibling to its left. The word is then scored, as in the Yngve measure, by summing the scores on the links along the path. Each non-terminal node in the path contributes a score of 1, except for sentence nodes and sentence-complement nodes,[3] which score 1.5 rather than 1. Thus embedded clauses contribute more to the complexity measure than other embedded categories, as an explicit acknowledgment of sentence embeddings as a source of syntactic complexity.

As with the Yngve score, we can calculate the total and the mean of these word scores. In contrast to the maximum score calculated for the Yngve measure, Frazier proposed summing the word scores for each 3-word sequence in the sentence, then taking the maximum of these sums as a measure of highly-localized concentrations of grammatical constituents. For the example in Figure 2, the maximum is 2.5, the maximum 3-word sum is 5.5, and the total is 7.5, yielding a mean of $\frac{15}{16}$.

3.3 Dependency distance

Rather than examining the tree structure itself, one might also extract measures from lexical dependency structures. These dependencies can be derived from the tree using standard rules for establishing head children for constituents, originally at-

[2] An exception is made for empty subject NPs, in which case the succeeding verb receives an additional score of 1 (for the deleted NP), and its path continues up the tree. Empty NPs are annotated in our manual parse trees but not in the automatic parses, which may result in a small disagreement in the Frazier scores for manual and automatic trees.

[3] Every non-terminal node beginning with an S, including SQ and SINV, were counted as sentence nodes. Sequences of sentence nodes, i.e. an SBAR appearing directly under an S node, were only counted as a single sentence node and thus only contributed to the score once.

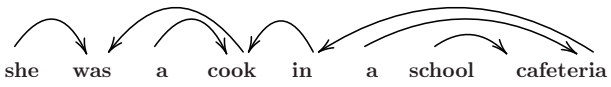


Figure 3: Dependency graph for the example string.

tributed to Magerman (1995), to percolate lexical heads up the tree. Figure 3 shows the dependency graph that results from this head percolation approach, where each link in the graph represents a dependency relation from the modifier to the head. For example, conventional head percolation rules specify the VP as the head of the S, so 'was', as the head of the VP, is thus the lexical head of the entire sentence. The lexical heads of the other children of the S node are called modifiers of the head of the S node; thus, since 'she' is the head of the subject NP, there is a dependency relation between 'she' and 'was'.

Lin (1996) argued for the use of this sort of dependency structure to measure the difficulty in processing, given the memory overhead of very long distance dependencies. Both Lin (1996) and Gibson (1998) showed that human performance on sentence processing tasks could be predicted with measures of this sort. While details may differ – e.g., how to measure distance, what counts as a dependency – we can make use of the general approach given Treebank style parses and head percolation, resulting in graphs of the sort in Figure 3. For the current paper, we count the distance between words for each dependency link. For Figure 3, there are 7 dependency links, a distance total of 11, and a mean of $1\frac{4}{7}$.

3.4 Developmental level (D-Level)

D-Level defines eight levels of sentence complexity, from 0-7, based on the development of complex sentences in normal-development children. Each level is defined by the presence of specific grammatical constructions (Rosenberg and Abbeduto, 1987); we follow Cheung and Kemper (1992) in assigning scores equivalent to the defined level of complexity. A score of zero corresponds to simple, single-clause sentences; embedded infinitival clauses get a score of 1 (*She needs to pay the rent*); conjoined clauses (*She worked all day and worried all night*), compound subjects (*The woman and her four children had not eaten for two days*), and wh-predicate complements score 2. Object noun phrase relative clauses or complements score 3 (*The police caught the man who robbed the woman*), whereas the same constructs in subject noun phrases score 5 (*The woman who worked in the cafeteria was robbed*). Gerundive complements and comparatives (*They were hungrier than her*) receive a score of 4; subordinating conjunctions (*if, before, as soon as*) score 6. Finally, a score of 7 is used as a catch-all category for sentences containing more than one of any of these grammatical constructions.

3.5 POS-tag sequence cross entropy

One possible approach for detecting rich syntactic structure is to look for infrequent or surprising combinations of parts-of-speech (POS). We can measure this over an utterance by building a simple bi-gram model over POS tags, then measuring the cross entropy of each utterance.[4]

Given a bi-gram model over POS-tags, we can calculate the probability of the sequence as a whole. Let τ_i be the POS-tag of word w_i in a sequence of words $w_1 \ldots w_n$, and assume that τ_0 is a special start symbol, and that τ_{n+1} is a special stop symbol. Then the probability of the POS-tag sequence is

$$\mathrm{P}(\tau_1 \ldots \tau_n) = \prod_{i=1}^{n+1} \mathrm{P}(\tau_i \mid \tau_{i-1}) \quad (1)$$

The cross entropy is then calculated as

$$H(\tau_1 \ldots \tau_n) = -\frac{1}{n} \log \mathrm{P}(\tau_1 \ldots \tau_n) \quad (2)$$

With this formulation, this basically boils down to the mean negative log probability of each tag given the previous tag.

4 Data

4.1 Subjects

We collected audio recordings of 55 neuropsychological examinations administered at the Layton Aging & Alzheimer's Disease Center, an NIA-funded Alzheimer's center for research at OHSU. For this study, we partitioned subjects into two groups: those who were assigned a Clinical Dementia Rating (CDR) of 0 (healthy) and those who were assigned a CDR of 0.5 (Mild Cognitive Impairment; MCI). The CDR (Morris, 1993) is assigned with access to clinical and cognitive test information, independent of performance on the battery of neuropsychological tests used for this research study, and has been shown to have high expert inter-annotator reliability (Morris et al., 1997).

[4]For each test domain, we used cross-validation techniques to build POS-tag bi-gram models and evaluate with them in that domain.

	CDR = 0 (n=29)		CDR = 0.5 (n=18)		
Measure	M	SD	M	SD	$t(45)$
Age	88.1	9.0	91.9	4.4	-1.65
Education (Y)	15.0	2.2	14.3	2.8	1.04
MMSE	28.4	1.4	25.9	2.6	4.29^{***}
Word List (A)	20.0	4.0	15.4	3.3	4.06^{***}
Word List (R)	6.8	2.0	3.9	1.7	5.12^{***}
Wechsler LM I	17.2	4.0	10.9	4.2	5.20^{***}
Wechsler LM II	15.8	4.3	9.5	5.4	4.45^{***}
Cat.Fluency (A)	17.2	4.1	13.9	4.2	2.59^{*}
Cat.Fluency (V)	12.8	4.5	9.6	3.6	2.57^{*}
Digits (F)	6.6	1.4	6.1	1.2	1.11
Digits (B)	4.7	1.0	4.7	1.1	-0.04

Table 1: Neuropsychological test results for subjects. $^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Of the collected recordings, three subjects were recorded twice; for the current study only one recording was used for each subject. Three subjects were assigned a CDR of 1.0 and were excluded from the study; two further subjects were excluded for errors in the recording that resulted in missing audio. Of the remaining 47 subjects, 29 had CDR = 0, and 18 had CDR = 0.5.

4.2 Neuropsychological tests

Table 1 presents means and standard deviations for age, years of education and the manually-calculated scores of a number of standard neuropsychological tests that were administered during the recorded session. These tests include: the Mini Mental State Examination (MMSE); the CERAD Word List Acquisition (A) and Recall (R) tests; the Wechsler Logical Memory (LM) I (immediate) and II (delayed) narrative recall tests; Category Fluency, Animals (A) and Vegetables (V); and Digit Span (WAIS-R) forward (F) and backward (B).

The Wechsler Logical Memory I/II tests are the basis of our study on syntactic complexity measures. The original narrative is a short, 3 sentence story:

> Anna Thompson of South Boston, employed as a cook in a school cafeteria, reported at the police station that she had been held up on State Street the night before and robbed of fifty-six dollars. She had four small children, the rent was due, and they had not eaten for two days. The police, touched by the woman's story, took up a collection for her.

Subjects are asked to re-tell this story immediately after it is told to them (LM I), as well as after approximately 30 minutes of unrelated activities (LM II). We transcribed each retelling, and manually annotated syntactic parse trees according to the Penn Treebank annotation guidelines. Algorithms for automatically extracting syntactic complexity markers from parse trees were written to accept either manually annotated trees or trees output from an automatic parser, to demonstrate the plausibility of using automatically generated parse trees.

System	LR	LP	F-measure
Out-of-domain (WSJ)	77.7	80.1	78.9
Out-of-domain (SWBD)	84.0	86.2	85.1
Domain adapted from SWBD	87.9	88.3	88.1

Table 2: Parser accuracy on Wechsler Logical Memory responses using just out-of-domain data (either from the Wall St. Journal (WSJ) or Switchboard (SWBD) treebanks) versus using a domain adapted system.

4.3 Parsing

For automatic parsing, we made use of the well-known Charniak parser (Charniak, 2000). Following best practices (Charniak and Johnson, 2001), we removed sequences covered by EDITED nodes in the tree from the strings prior to parsing. For this paper, EDITED nodes were identified from the manual parse, not automatically. Table 2 shows parsing accuracy of our annotated retellings under three parsing model training conditions: 1) trained on approximately 1 million words of Wall St. Journal (WSJ) text; 2) trained on approximately 1 million words of Switchboard (SWBD) corpus telephone conversations; and 3) using domain adaptation techniques starting from the SWBD Treebank. The SWBD out-of-domain system reaches quite respectable accuracies, and domain adaptation achieves 3 percent absolute improvement over that.

For domain adaptation, we used MAP adaptation techniques (Bacchiani et al., 2006) via cross-validation over the entire set of retellings. For each subject, we trained a model using the SWBD treebank as the out-of-domain treebank, and the retellings of the other 46 subjects as in-domain training. We used a count merging approach, with the in-domain counts scaled by 1000 relative to the out-of-domain counts. See Bacchiani et al. (2006) for more information on stochastic grammar adaptation using these techniques.

5 Experimental results

5.1 Correlations

Our first set of experimental results regard correlations between measures. Table 3 shows results for five of our measures over all three treebanks that we have been considering: Penn WSJ Treebank, Penn SWBD Treebank, and the Wechsler LM retellings. The correlations along the diagonal are between the same measure when extracted from manually annotated trees and when extracted from automatic

	Penn WSJ Treebank					Penn SWBD Treebank					Wechsler LM Retellings				
	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)
(a) Frazier	0.89					0.96					0.94				
(b) Yngve	-0.31	0.96				-0.72	0.96				-0.69	0.95			
(c) Tree nodes	0.91	-0.16	0.92			0.58	-0.06	0.93			0.93	-0.48	0.85		
(d) Dep len	-0.29	0.75	-0.13	0.93		-0.74	0.97	-0.08	0.96		-0.72	0.96	-0.51	0.96	
(e) Cross Ent	0.17	0.18	0.15	0.19	0.93	-0.55	0.76	0.09	0.76	0.98	-0.13	0.45	0.05	0.41	0.97

Table 3: Correlation matrices for several measures on an utterance-by-utterance basis. Correlations along the diagonal are between the manual measures and the measures when automatically parsed. All other correlations are between measures when derived from manual parse trees.

parses. All other correlations are between measures derived from manual trees. All correlations are taken per utterance.

From this table, we can see that all of the measures derived from automatic parses have a high correlation with the manually derived measures, indicating that they may preserve any discriminative utility of these markers. Interestingly, the number of nodes in the tree per word tends to correlate well with the Frazier score, while the dependency length tends to correlate well with the Yngve score. Cross entropy correlates with Yngve and dependency length for the SWBD and Wechsler treebanks, but not for the WSJ treebank.

5.2 Manually derived measures

Table 4 presents means and standard deviations for measures derived from the LM I and LM II retellings, along with the t-value and level of significance. The first three measures presented in the table are available without syntactic annotation: total number of words, total number of utterances, and words per utterance in the retelling. None of these three measures on either retelling show statistically significant differences between the groups.

The first measure to rely upon syntactic annotations is words per clause. The number of clauses are automatically extracted from the parses by counting the number of S nodes in the tree.[5] Normalizing the number of words by the number of clauses rather than the number of utterances (as in words per utterance) results in statistically significant differences between the groups for LM I though not for LM II.

The other measures are as described in Section 3. Interestingly, Frazier score per word, the number of tree nodes per word, and POS-tag cross entropy all show a significant negative t-value on the LM I retellings, meaning the CDR 0.5 subjects had significantly higher scores than the CDR 0 subjects for these measures on this task. These measures showed no significant difference on the LM II retellings.

The Yngve score per word and the dependency length per word showed no significant difference on LM I retellings but a statistically significant difference on LM II, with the expected outcome of higher scores for the CDR 0 subjects. The D-Level measure showed no significant differences.

5.3 Automatically derived measures

In addition to manual-parse derived measures, Table 4 also presents the same measures when automatic, rather than manual, parses are used. Given the relatively high quality of the automatic parses, most of the means and standard deviations are quite close, and all of the patterns observed in the upper half of Table 4 are preserved, except that the Yngve score per word no longer shows a statistically significant difference for the LM II retelling.

5.4 Left-corner trees

For the tree-based complexity metrics (Frazier and Yngve), we also investigated alternative implementations that make use of the left-corner transformation (Rosenkrantz and Lewis II, 1970) of the tree from which the measures were extracted. This transformation is widely known for removing left-recursion from a context-free grammar, and it changes the tree shape by transforming left-branching structures into right-branching structures, while leaving center-embedded structures center-embedded. This property led Resnik (1992) to propose left-corner processing as a plausible mechanism for human sentence processing, since it is precisely these center-embedded structures, and not the left- or right-branching structures, that are problematic for humans to process.

Table 5 presents results using either manually annotated trees or automatic parses to extract the Yngve and Frazier measures after a left-corner transform has been applied to the tree. The Frazier scores are very similar to those without the left-

[5]For coordinated S nodes, the root of the coordination, which in Penn Treebank style annotation also has an S label, does not count as an additional clause.

Measure	Logical Memory I					Logical Memory II				
	CDR = 0		CDR = 0.5			CDR = 0		CDR = 0.5		
	M	SD	M	SD	$t(45)$	M	SD	M	SD	$t(45)$
Total words in retelling	71.0	26.0	58.1	31.9	1.49	70.6	21.5	58.5	36.7	1.43
Total utterances in retelling	8.86	4.16	7.72	3.28	0.99	8.17	2.77	7.06	4.86	1.01
Words per utterance in retelling	8.57	2.44	7.78	3.67	0.89	9.16	3.06	7.82	4.76	1.18
Manually extracted: Words per clause	6.33	1.39	5.25	1.25	$2.68^{\star}$	6.12	1.20	5.48	3.37	0.93
Frazier score per word	1.19	0.09	1.26	0.11	$-2.68^{\star}$	1.19	0.09	1.13	0.43	0.67
Tree nodes per word	1.96	0.07	2.01	0.10	$-2.08^{\star}$	1.96	0.07	1.80	0.66	1.36
Yngve score per word	1.44	0.23	1.39	0.30	0.61	1.53	0.27	1.26	0.62	$2.01^{\star}$
Dependency length per word	1.54	0.25	1.47	0.27	0.90	1.63	0.30	1.34	0.60	$2.19^{\star}$
POS-tag Cross Entropy	1.83	0.16	1.96	0.26	$-2.18^{\star}$	1.93	0.14	1.86	0.59	0.54
D-Level	1.07	0.75	1.03	1.23	0.14	1.23	0.81	1.68	1.41	-1.42
Auto extracted: Words per clause	6.42	1.53	5.10	1.16	$3.13^{\star\star}$	6.04	1.25	5.61	3.67	0.59
Frazier score per word	1.16	0.10	1.24	0.10	$-2.92^{\star\star}$	1.15	0.10	1.09	0.41	0.69
Tree nodes per word	1.96	0.07	2.03	0.10	$-2.55^{\star}$	1.96	0.08	1.79	0.66	1.38
Yngve score per word	1.41	0.23	1.37	0.29	0.54	1.50	0.27	1.28	0.64	1.70
Dependency length per word	1.51	0.25	1.47	0.28	0.54	1.61	0.28	1.35	0.61	$2.04^{\star}$
POS-tag Cross Entropy	1.83	0.17	1.96	0.26	$-2.12^{\star}$	1.92	0.14	1.86	0.58	0.53
D-Level	1.09	0.73	1.11	1.20	-0.08	1.28	0.77	1.61	1.22	-1.15

Table 4: Syntactic complexity measure group differences when measures are derived from either manual or automatic parse trees. $^{\star\star}p < 0.01$; $^{\star}p < 0.05$

corner transform, while the Yngve scores are reduced across the board. With the left-corner transformed tree, the automatically derived Yngve measure retains the statistically significant difference shown by the manually derived measure.

6 Discussion and future directions

The results presented in the last section demonstrate that NLP techniques applied to clinically elicited spoken language samples can be used to automatically derive measures that may be useful for discriminating between healthy and MCI subjects. In addition, we see that different measures show different patterns when applied to these language samples, with Frazier scores and tree nodes per word giving quite different results than Yngve scores and dependency length. It would thus appear that, for Penn Treebank style annotations at least, these measures are quite complementary.

There are two surprising aspects of these results: the significantly higher means of three measures on LM I samples for MCI subjects, and the fact that one set of measures show significant differences on LM I while another shows differences on LM II. We do not have definitive explanations for these phenomena, but we can speculate about why such results were obtained.

First, there is an important difference between the manner of elicitation for LM I versus LM II. LM I is an immediate recall, so there will likely be, for unimpaired subjects, much higher verbatim recall of the story than in the delayed recall of LM II. For the MCI group, which exhibits memory impairment, there will be little in the way of verbatim recall, and potentially much more in the way of spoken language phenomena such as filled pauses, parentheticals and off-topic utterances. This may account for the higher Frazier score per word for the MCI group on LM I. Such differences will likely be lessened in the delayed recall.

Second, the Frazier and Yngve metrics differ in how they score long, flat phrases, such as typical base NPs. Consider the ternary NP in Figure 1. The first word in that NP ('a') receives an Yngve score of 2, but a Frazier score of only 1 (Figure 2), while the second word in the NP receives an Yngve score of 1 and a Frazier score of 0. For a flat NP with 5 children, that difference would be 4 to 1 for the first child, 3 to 0 for the second child, and so forth. This difference in scoring relatively common syntactic constructions, even those which may not affect human memory load, may account for such different scores achieved with these different measures.

In summary, we have demonstrated an important clinical use for NLP techniques, where automatic syntactic annotation provides sufficiently accurate parse trees for use in automatic extraction of syntactic complexity measures. Different syntactic complexity measures appear to be measuring quite complementary characteristics of the retellings, yielding statistically significant differences from both immediate and delayed retellings.

There are quite a number of questions that we will

	Logical Memory I					Logical Memory II				
	CDR = 0		CDR = 0.5			CDR = 0		CDR = 0.5		
Measure	M	SD	M	SD	$t(45)$	M	SD	M	SD	$t(45)$
Manually extracted: Left-corner Frazier	1.20	0.10	1.28	0.12	$-2.60^{\star}$	1.20	0.11	1.18	0.45	0.29
Left-corner Yngve	1.33	0.20	1.25	0.23	1.20	1.37	0.21	1.14	0.52	$2.14^{\star}$
Auto extracted: Left-corner Frazier	1.16	0.10	1.27	0.13	$-3.02^{\star\star}$	1.15	0.11	1.10	0.42	0.64
Left-corner Yngve	1.31	0.19	1.23	0.21	1.33	1.36	0.21	1.13	0.51	$2.11^{\star}$

Table 5: Syntactic complexity measure group differences when measures are derived from left-corner parse trees. $^{\star\star}p < 0.01$; $^{\star}p < 0.05$

continue to pursue. Most importantly, we will continue to examine this data, to try to determine what characteristics of the spoken language are leading to the unexpected patterns in the results. In addition, we will begin to explore composite measures, such as differences in measures between LM I and LM II, which promise to better capture some of the patterns we have observed. Ultimately, we would like to build classifiers making use of a range of measures as features, although in order to demonstrate statistically significant differences between classifiers, we will need much more data than we currently have. Eventually, longitudinal tracking of subjects may be the best application of such measures on clinically elicited spoken language samples.

Acknowledgments

This research was supported in part by NSF Grant #IIS-0447214 and pilot grants from the Oregon Center for Aging & Technology (ORCATECH, NIH #1P30AG024978-01) and the Oregon Partnership for Alzheimer's Research. Also, the third author of this paper was supported under an NSF Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the NSF. Thanks to Jeff Kaye, John-Paul Hosom, Jan van Santen, Tracy Zitzelberger, Jessica Payne-Murphy and Robin Guariglia for help with the project.

References

M. Bacchiani, M. Riley, B. Roark, and R. Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

E. Charniak and M. Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proceedings of the 2nd Conference of the North American Chapter of the ACL.*

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the ACL*, pages 132–139.

H. Cheung and S. Kemper. 1992. Competing complexity metrics and adults' production of complex sentences. *Applied Psycholinguistics*, 13:53–76.

L. Frazier. 1985. Syntactic complexity. In D.R. Dowty, L. Karttunen, and A.M. Zwicky, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, UK.

E. Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

S. Kemper, E. LaBarge, F.R. Ferraro, H. Cheung, H. Cheung, and M. Storandt. 1993. On the preservation of syntax in Alzheimer's disease. *Archives of Neurology*, 50:81–86.

D. Lin. 1996. On structural complexity. In *Proceedings of COLING-96*.

K. Lyons, S. Kemper, E. LaBarge, F.R. Ferraro, D. Balota, and M. Storandt. 1994. Oral language and Alzheimer's disease: A reduction in syntactic complexity. *Aging and Cognition*, 1(4):271–281.

D.M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the ACL*, pages 276–283.

M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

J.F. Miller and R.S. Chapman. 1981. The relation between age and mean length of utterance in morphemes. *Journal of Speech and Hearing Research*, 24:154–161.

J. Morris, C. Ernesto, K. Schafer, M. Coats, S. Leon, M. Sano, L. Thal, and P. Woodbury. 1997. Clinical dementia rating training and reliability in multicenter studies: The Alzheimer's disease cooperative study experience. *Neurology*, 48(6):1508–1510.

J. Morris. 1993. The clinical dementia rating (CDR): Current version and scoring rules. *Neurology*, 43:2412–2414.

P. Resnik. 1992. Left-corner parsing and psychological plausibility. In *Proceedings of COLING-92*, pages 191–197.

B. Roark, J.P. Hosom, M. Mitchell, and J.A. Kaye. 2007. Automatically derived spoken language markers for detecting mild cognitive impairment. In *Proceedings of the 2nd International Conference on Technology and Aging (ICTA)*.

S. Rosenberg and L. Abbeduto. 1987. Indicators of linguistic competence in the peer group conversational behavior of mildly retarded adults. *Applied Psycholinguistics*, 8:19–32.

S.J. Rosenkrantz and P.M. Lewis II. 1970. Deterministic left corner parsing. In *IEEE Conference Record of the 11th Annual Symposium on Switching and Automata*, pages 139–152.

K. Sagae, A. Lavie, and B. MacWhinney. 2005. Automatic measurement of syntactic development in child langugage. In *Proceedings of the 43rd Annual Meeting of the ACL.*

H.S. Scarborough. 1990. Index of productive syntax. *Applied Psycholinguistics*, 11:1–22.

S. Singh, R.S. Bucks, and J.M. Cuerden. 2001. Evaluation of an objective technique for analysing temporal variables in DAT spontaneous speech. *Aphasiology*, 15(6):571–584.

V.H. Yngve. 1960. A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104:444–466.

Determining the Syntactic Structure of Medical Terms in Clinical Notes

Bridget T. McInnes
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN, 55455
bthomson@cs.umn.edu

Ted Pedersen
Dept. of Computer Science
University of Minnesota Duluth
Duluth, MN, 55812
tpederse@d.umn.edu

Serguei V. Pakhomov
Dept. of Pharmaceutical Care and Health Systems Center for Health Informatics
University of Minnesota
Minneapolis, MN, 55455
pakh0002@umn.edu

Abstract

This paper demonstrates a method for determining the syntactic structure of medical terms. We use a model-fitting method based on the Log Likelihood Ratio to classify three-word medical terms as right or left-branching. We validate this method by computing the agreement between the classification produced by the method and manually annotated classifications. The results show an agreement of 75% - 83%. This method may be used effectively to enable a wide range of applications that depend on the semantic interpretation of medical terms including automatic mapping of terms to standardized vocabularies and induction of terminologies from unstructured medical text.

1 Introduction

Most medical concepts are expressed via a domain specific terminology that can either be explicitly agreed upon or extracted empirically from domain specific text. Regardless of how it is constructed, a terminology serves as a foundation for information encoding, processing and exchange in a specialized sub-language such as medicine. Concepts in the medical domain are encoded through a variety of linguistic forms, the most typical and widely accepted is the noun phrase (NP). In some even further specialized subdomains within medicine, such as nursing and surgery, an argument can be made that some concepts are represented by an entire predication rather than encapsulated within a single nominalized expression. For example, in order to describe someone's ability to lift objects 5 pounds or heavier above their head, it may be necessary to use a term consisting of a predicate such as [LIFT] and a set of arguments corresponding to various thematic roles such as <PATIENT> and <PATH> (Ruggieri et al., 2004). In this paper, we address typical medical terms encoded as noun phrases (NPs) that are often structurally ambiguous, as in Example 1, and discuss a case for extending the proposed method to non-nominalized terms as well.

$$\text{small}_1\ \text{bowel}_2\ \text{obstruction}_3 \quad (1)$$

The NP in Example 1 can have at least two interpretations depending on the syntactic analysis:

$$[[\text{small}_1\ \text{bowel}_2]\ \text{obstruction}_3] \quad (2)$$

$$[\text{small}_1\ [\text{bowel}_2\ \text{obstruction}_3]] \quad (3)$$

The term in Example 2 denotes an obstruction in the small bowel, which is a diagnosable disorder; whereas, the term in Example 3 refers to a small unspecified obstruction in the bowel.

Unlike the truly ambiguous general English cases such as the classical "American History Professor" where the appropriate interpretation depends on the context, medical terms, such as in Example 1, tend to have only one appropriate interpretation. The context, in this case, is the discourse domain of medicine. From the standpoint of the English language, the interpretation that follows from Example 3 is certainly plausible, but unlikely in the context of a medical term. The syntax of a term only shows

BioNLP 2007: Biological, translational, and clinical language processing, pages 9–16,
Prague, June 2007.

what interpretations are possible without restricting them to any particular one. From the syntactic analysis, we know that the term in Example 1 has the potential for being ambiguous; however, we also know that it does have an intended interpretation by virtue of being an entry term in a standardized terminology with a unique identifier anchoring its meaning. What we do not know is which syntactic structure generated that interpretation. Being able to determine the structure consistent with the intended interpretation of a clinical term can improve the analysis of unrestricted medical text and subsequently improve the accuracy of Natural Language Processing (NLP) tasks that depend on semantic interpretation.

To address this problem, we propose to use a model-fitting method which utilizes an existing statistical measure, the Log Likelihood Ratio. We validate the application of this method on a corpus of manually annotated noun-phrase-based medical terms. First, we present previous work on structural ambiguity resolution. Second, we describe the Log Likelihood Ratio and then its application to determining the structure of medical terms. Third, we describe the training corpus and discuss the compilation of a test set of medical terms and human expert annotation of those terms. Last, we present the results of a preliminary validation of the method and discuss several possible future directions.

2 Previous Work

The problem of resolving structural ambiguity has been previously addressed in the computational linguistics literature. There are multiple approaches ranging from purely statistical (Ratnaparkhi, 1998), to hybrid approaches that take into account the lexical semantics of the verb (Hindle and Rooth, 1993), to corpus-based, which is the approach discussed in this paper. (Marcus, 1980) presents an early example of a corpus-based approach to syntactic ambiguity resolution. One type of structural ambiguity that has received much attention has to do with nominal compounds as seen in the work of (Resnik, 1993), (Resnik and Hearst, 1993), (Pustejovsky et al., 1993), and (Lauer, 1995).

(Lauer, 1995) points out that the existing approaches to resolving the ambiguity of noun phrases fall roughly into two camps: adjacency and dependency. The proponents of the adjacency model ((Liberman and Sproat, 1992), (Resnik, 1993) and (Pustejovsky et al., 1993)) argue that, given a three word noun phrase XYZ, there are two possible analyzes [[XY]Z] and [X[YZ]]. The correct analysis is chosen based on the "acceptability" of the adjacent bigrams A[XY] and A[YZ]. If A[XY] is more acceptable than A[YZ], then the left-branching analysis [[XY]Z] is preferred.

(Lauer and Dras, 1994) and (Lauer, 1995) address the issue of structural ambiguity by developing a dependency model where instead of computing the acceptability of A[YZ] one would compute the acceptability of A[XZ]. (Lauer, 1995) argues that the dependency model is not only more intuitive than the adjacency model, but also yields better results. (Lapata and Keller, 2004) results also support this assertion.

The difference between the approaches within the two models is the computation of acceptability. Proposals for computing acceptability (or preference) include raw frequency counts ((Evans and Zhai, 1996) and (Lapata and Keller, 2004)), Latent Semantic Indexing ((Buckeridge and Sutcliffe, 2002)) and statistical measures of association ((Lapata et al., 1999) and (Nakov and Hearst, 2005)).

One of the main problems with using frequency counts or statistical methods for structural ambiguity resolution is the sparseness of data; however, (Resnik and Hearst, 1993) used conceptual associations (associations between groups of terms deemed to form conceptual units) in order to alleviate this problem. (Lapata and Keller, 2004) use the document counts returned by WWW search engines. (Nakov and Hearst, 2005) use the χ^2 measure based on statistics obtained from WWW search engines to compute values to determine acceptability of a syntactic analysis for nominal compounds. This method is tested using a set of general English nominal compounds developed by (Lauer, 1995) as well as a set of nominal compounds extracted from MEDLINE abstracts.

The novel contribution of our study is in demonstrating and validating a corpus-based method for determining the syntactic structure of medical terms that relies on using the statistical measure of association, the Log Likelihood Ratio, described in the following section.

3 Log Likelihood Ratio

The Log Likelihood Ratio (G^2) is a "goodness of fit" statistic first proposed by (Wilks, 1938) to test if a given piece of data is a sample from a set of data with a specific distribution described by a hypothesized model. It was later applied by (Dunning, 1993) as a way to determine if a sequence of N words (N-gram) came from an independently distributed sample.

(Pedersen et al., 1996) pointed out that there exists theoretical assumptions underlying the G^2 measure that were being violated therefore making them unreliable for significance testing. (Moore, 2004) provided additional evidence that although G^2 may not be useful for determining the significance of an event, its near equivalence to mutual information makes it an appropriate measure of word association. (McInnes, 2004) applied G^2 to the task of extracting three and four word collocations from raw text.

G^2, formally defined for trigrams in Equation 4, compares the observed frequency counts with the counts that would be expected if the words in the trigram (3-gram; a sequence of three words) corresponded to the hypothesized model.

$$G^2 = 2 * \sum_{x,y,z} n_{xyz} * log(\frac{n_{xyz}}{m_{xyz}}) \qquad (4)$$

The parameter n_{xyz} is the observed frequency of the trigram where x, y, and z respectively represent the occurrence of the first, second and third words in the trigram. The variable m_{xyz} is the expected frequency of the trigram which is calculated based on the hypothesized model. This calculation varies depending on the model used. Often the hypothesized model used is the independence model which assumes that the words in the trigram occur together by chance. The calculation of the expected values based on this model is as follows:

$$m_{xyz} = n_{x++} * n_{+y+} * n_{++z}/n_{+++} \qquad (5)$$

The parameter, n_{+++}, is the total number of trigrams that exist in the training data, and n_{x++}, n_{+y+}, and n_{++z} are the individual marginal counts of seeing words x, y, and z in their respective positions in a trigram. A G^2 score reflects the degree to which the observed and expected values diverge. A G^2 score of zero implies that the observed values are equal to the expected and the trigram is represented perfectly by the hypothesized model. Hence, we would say that the data 'fits' the model. Therefore, the higher the G^2 score, the less likely the words in the trigram are represented by the hypothesized model.

4 Methods

4.1 Applying Log Likelihood to Structural Disambiguation

The independence model is the only hypothesized model used for bigrams (2-gram; a sequence of two words). As the number of words in an N-gram grows, the number of hypothesized models also grows. The expected values for a trigram can be based on four models. The first model is the independence model discussed above. The second is the model based on the probability that the first word and the second word in the trigram are dependent and independent of the third word. The third model is based on the probability that the second and third words are dependent and independent of the first word. The last model is based on the probability that the first and third words are dependent and independent of the second word. Table 1 shows the different models for the trigram XYZ.

Table 1: Models for the trigram XYZ

Model 1	P(XYZ) / P(X) P(Y) P(Z)
Model 2	P(XYZ) / P(XY) P(Z)
Model 3	P(XYZ) / P(X) / P(YZ)
Model 4	P(XYZ) / P(XZ) P(Y)

Slightly different formulas are used to calculate the expected values for the different hypothesized models. The expected values for Model 1 (the independence model) are given above in Equation 5. The calculation of expected values for Model 2, 3, 4 are seen in Equations 6, 7, 8 respectively.

$$m_{xyz} = n_{xy+} * n_{++z}/n_{+++} \qquad (6)$$

$$m_{xyz} = n_{x++} * n_{+yz}/n_{+++} \qquad (7)$$

$$m_{xyz} = n_{x+z} * n_{+y+}/n_{+++} \qquad (8)$$

The parameter n_{xy+} is the number of times words x and y occur in their respective positions, n_{+yz} is

the number of times words y and z occur in their respective positions and n_{x+z} is the number of times that words x and z occur in their respective positions in the trigram.

The hypothesized models result in different expected values which results in a different G^2 score. A G^2 score of zero implies that the data are perfectly represented by the hypothesized model and the observed values are equal to the expected. Therefore, the model that returns the lowest score for a given trigram is the model that best represents the structure of that trigram, and hence, best 'fits' the trigram. For example, Table 2 shows the scores returned for each of the four hypothesized models for the trigram "small bowel obstruction".

Table 2: Example for the term "small bowel obstruction"

Model	G^2 score	Model	G^2 score
Model 1	11,635.45	Model 2	**5,169.81**
Model 3	8,532.90	Model 4	7,249.90

The smallest G^2 score is returned by Model 2 which is based on the first and second words being dependent and independent of the third. Based on the data, Model 2 best represents or 'fits' the trigram, "small bowel obstruction". In this particular case that happens to be the correct analysis.

The frequency counts and G^2 scores for each model were obtained using the N-gram Statistics Package [1] (Banerjee and Pedersen, 2003).

4.2 Data

The data for this study was collected from two sources: the Mayo Clinic clinical notes and SNOMED-CT terminology (Stearns et al., 2001).

4.2.1 Clinical Notes

The corpus used in this study consists of over 100,000 clinical notes covering a variety of major medical specialties at the Mayo Clinic. These notes document each patient-physician contact and are typically dictated over the telephone. They range in length from a few lines to several pages of text and represent a quasi-spontaneous discourse where the dictations are made partly from notes and partly from memory. At the Mayo Clinic, the dictations are transcribed by trained personnel and are stored in the patient's chart electronically.

4.2.2 SNOMED-CT

SNOMED-CT (Systematized Nomenclature of Medicine, Clinical Terminology) is an ontological resource produced by the College of American Pathologists and distributed as part of the Unified Medical Language System[2] (UMLS) Metathesaurus maintained by the National Library of Medicine. SNOMED-CT is the single largest source of clinical terms in the UMLS and as such lends itself well to the analysis of terms found in clinical reports.

SNOMED-CT is used for many applications including indexing electronic medical records, ICU monitoring, clinical decision support, clinical trials, computerized physician order entry, disease surveillance, image indexing and consumer health information services. The version of SNOMED-CT used in this study consists of more than 361,800 unique concepts with over 975,000 descriptions (entry terms) (SNOMED-CT Fact Sheet, 2004).

4.3 Testset of Three Word Terms

We used SNOMED-CT to compile a list of terms in order to develop a test set to validate the G^2 method. The test set was created by extracting all trigrams from the corpus of clinical notes and all three word terms found in SNOMED-CT. The intersection of the SNOMED-CT terms and the trigrams found in the clinical notes was further restricted to include only simple noun phrases that consist of a head noun modified with a set of other nominal or adjectival elements including adjectives and present and past participles. Adverbial modification of adjectives was also permitted (e.g. "partially edentulous maxilla"). Noun phrases with nested prepositional phrases such as "fear of flying" as well as three word terms that are not noun phrases such as "does not eat" or "unable to walk" were excluded from the test set. The resulting test set contains 710 items.

The intended interpretation of each three word term (trigram) was determined by arriving at a

[1] http://www.d.umn.edu/ tpederse/nsp.html

[2] Unified Medical Language System is a compendium of over 130 controlled medical vocabularies encompassing over one million concepts.

consensus between two medical index experts (kappa=0.704). These experts have over ten years of experience with classifying medical diagnoses and are highly qualified to carry out the task of determining the intended syntactic structure of a clinical term.

Table 3: Four Types of Syntactic Structures of Trigram Terms

left-branching ((XY)Z):
[[urinary tract] infection]
[[right sided] weakness]
right-branching (X(YZ)):
[chronic [back pain]]
[low [blood pressure]]
non-branching ((X)(Y)(Z)):
[[follicular][thyroid][carcinoma]]
[[serum][dioxin][level]]
monolithic (XYZ):
[difficulty finding words]
[serous otitis media]

In the process of annotating the test set of trigrams, four types of terms emerged (Table 3). The first two types are left and right-branching where the left-branching phrases contain a left-adjoining group that modifies the head of the noun phrase. The right-branching phrases contain a right-adjoining group that forms the kernel or the head of the noun phrase and is modified by the remaining word on the left. The non-branching type is where the phrase contains a head noun that is independently modified by the other two words. For example, in "follicular thyroid carcinoma", the experts felt that "carcinoma" was modified by both "follicular" and "thyroid" independently, where the former denotes the type of cancer and the latter denotes its location. This intuition is reflected in some formal medical classification systems such as the Hospital International Classification of Disease Adaptation (HICDA) where cancers are typically classified with at least two categories - one for location and one for the type of malignancy. This type of pattern is rare. We were able to identify only six examples out of the 710 terms. The monolithic type captures the intuition that the terms function as a collocation and are not decomposable into subunits. For example, "leg length discrepancy" denotes a specific disorder where one leg is of a different length from the other. Various combinations of subunits within this term result in nonsensical expressions.

Table 4: Distribution of term types in the test set

Type	Count	%total
Left-branching	251	35.5
Right-branching	378	53.4
Non-branching	6	0.8
Monolithic	73	10.3
Total	708	100

Finally, there were two terms for which no consensus could be reached: "heart irregularly irregular" and "subacute combined degeneration". These cases were excluded from the final set. Table 4 shows the distribution of the four types of terms in the test set.

5 Evaluation

We hypothesize that general English typically has a specific syntactic structure in the medical domain, which provides a single semantic interpretation. The patterns observed in the set of 710 medical terms described in the previous section suggest that the G^2 method offers an intuitive way to determine the structure of a term that underlies its syntactic structure.

Table 5: G^2 Model Descriptions

left-branching	Model 2	[[XY] Z]
right-branching	Model 3	[X [YZ]]

The left and right-branching patterns roughly correspond to Models 2 and 3 in Table 5. Models 1 and 4 do not really correspond to any of the patterns we were able to identify in the set of terms. Model 1 would represent a term where words are completely independent of each other, which is an unlikely scenario given that we are working with terms whose composition is dependent by definition. This is not to say that in other applications (e.g., syntactic parsing) this model would not be relevant. Model 4 suggests dependence between the outer edges of a term and their independence from the

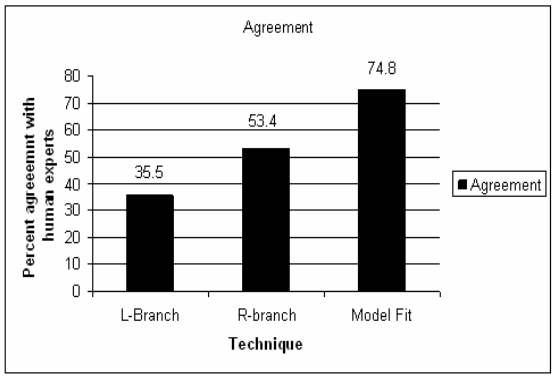


Figure 1: Comparison of the results with two baselines: L-branching and R-branching assumptions

middle word, which is not motivated from the standpoint of a traditional context free grammar which prohibits branch crossing. However, this model may be welcome in a dependency grammar paradigm.

One of the goals of this study is to test an application of the G^2 method trained on a corpus of medical data to distinguish between left and right-branching patterns. The method ought to suggest the most likely analysis for an NP-based medical term based on the empirical distribution of the term and its components. As part of the evaluation, we compute the G^2 scores for each of the terms in the test set, and picked the model with the lowest score to represent the structural pattern of the term. We compared these results with manually identified patterns. At this preliminary stage, we cast the problem of identifying the structure of a three word medical term as a binary classification task where a term is considered to be either left or right-branching, effectively forcing all terms to either be represented by either Model 2 or Model 3.

6 Results and Discussion

In order to validate the G^2 method for determining the structure of medical terms, we calculated the agreement between human experts' interpretation of the syntactic structure of the terms and the interpretation suggested by the G^2 method. The agreement was computed as the ratio of matching interpretations to the total number of terms being interpreted. We used two baselines, one established by assuming that each term is left-branching and the other by assuming that each term is right-branching. As is clear from Table 4, the left-branching baseline is 35.5% and the right-branching baseline is 53.4% meaning that if we simply assign left-branching pattern to each three word term, we would agree with human experts 35.5% of the time. The G^2 method correctly identifies 185 trigrams as being left-branching (Model 2) and 345 trigrams as being right-branching (Model 3). There are 116 right-branching trigrams incorrectly identified as left-branching, and 62 left-branching trigrams incorrectly identified as right- branching. Thus the method and the human experts agreed on 530 (75%) terms out of 708 (kappa=0.473), which is better than both baselines (Figure 1). We did not find any overlap between the terms that human experts annotated as non-branching and the terms whose corpus distribution can be represented by Model 4 ([[XZ]Y]). This is not surprising as this pattern is very rare. Most of the terms are represented by either Model 2 (left-branching) or Model 3 (right-branching). The monolithic terms that the human experts felt were not decomposable constitute 10% of all terms and may be handled through some other mechanism such as collocation extraction or dictionary lookup. Excluding monolithic terms from testing results in 83.5% overall agreement (kappa=0.664).

We observed that 53% of the terms in our test set are right-branching while only 35% are left-branching. (Resnik, 1993) found between 64% and 67% of nominal compounds to be left-branching and used that finding to establish a baseline for his experiments with structural ambiguity resolution. (Nakov and Hearst, 2005) also report a similar percentage (66.8%) of left-branching noun compounds. Our test set is not limited to nominal compounds, which may account for the fact that a slight majority of the terms are found to be right-branching as adjectival modification in English is typically located to the left of the head noun. This may also help explain the fact that the method tends to have higher agreement within the set of right-branching terms (85%) vs. left-branching (62%).

We also observed that many of the terms marked as monolithic by the experts are of Latin origin such as the term in Example 9 or describe the functional

status of a patient such as the term in Example 10.

$$\text{erythema}_1\ \text{ab}_2\ \text{igne}_3 \quad (9)$$

$$\text{difficulty}_1\ \text{swallowing}_2\ \text{solids}_3 \quad (10)$$

Example 10 merits further discussion as it illustrates another potential application of the method in the domain of functional status terminology. As was mentioned in the introduction, functional status terms may be be represented as a predication with a set of arguments. Such view of functional status terminology lends itself well to a frame-based representation of functional status terms in the context of a database such as FrameNet [3] or PropBank[4]. One of the challenging issues in representing functional status terminology in terms of frames is the distinction between the core predicate and the frame elements (Ruggieri et al., 2004). It is not always clear what lexical material should be part of the core predicate and what lexical material should be part of one or more arguments. Consider the term in Example 10 which represents a nominalized form of a predication. Conceivably, we could analyze this term as a frame shown in Example 11 where the predication consists of a predicate [DIFFICULTY] and two arguments. Alternatively, Example 12 presents a different analysis where the predicate is a specific kind of difficulty with a single argument.

[P:DIFFICULTY]
[ARG1:SWALLOWING$_{<\text{ACTIVITY}>}$] (11)
[ARG2:SOLIDS$_{<\text{PATIENT}>}$]

[P:SWALLOWING DIFFICULTY]
[ARG1: SOLIDS$_{<\text{PATIENT}>}$] (12)

The analysis dictates the shape of the frames and how the frames would fit into a network of frames. The G^2 method identifies Example 10 as left-branching (Model 2), which suggests that it would be possible to have a parent DIFFICULTY frame and a child CLIMBING DIFFICULTY that would inherit form its parent. An example where this is not possible is the term "difficulty staying asleep" where it would probably be nonsensical or at least impractical to have a predicate such as [STAYING DIFFICULTY]. It would be more intuitive to assign this term to the DIFFICULTY frame with a frame element whose lexical content is "staying asleep". The method appropriately identifies the term "difficulty staying asleep" as right-branching (Model 3) where the words "staying asleep" are grouped together. This is an example based on informal observations; however, it does suggest a utility in constructing frame-based representation of at least some clinical terms.

7 Limitations

The main limitation of the G^2 method is the exponential growth in the number of models to be evaluated with the growth in the length of the term. This limitation can be partly alleviated by either only considering adjacent models and limiting the length to 5-6 words, or using a forward or backward sequential search proposed by (Pedersen et al., 1997) for the problem of selecting models for the Word Sense Disambiguation task.

8 Conclusions and Future Work

This paper presented a simple but effective method based on G^2 to determine the internal structure of three-word noun phrase medical terms. The ability to determine the syntactic structure that gives rise to a particular semantic interpretation of a medical term may enable accurate mapping of unstructured medical text to standardized terminologies and nomenclatures. Future directions to improve the accuracy of our method include determining how other measures of association, such as dice coefficient and χ^2, perform on this task. We feel that there is a possibility that no single measure performs best over all types of terms. In that case, we plan to investigate incorporating the different measures into an ensemble-based algorithm.

We believe the model-fitting method is not limited to structural ambiguity resolution. This method could be applied to automatic term extraction and automatic text indexing of terms from a standardized vocabulary. More broadly, the principles of using distributional characteristics of word sequences derived from large corpora may be applied to unsupervised syntactic parsing.

[3] http://www.icsi.berkeley.edu/framenet/
[4] http://www.cis.upenn.edu/ ace/

Acknowledgments

We thank Barbara Abbott, Debra Albrecht and Pauline Funk for their contribution to annotating the test set and discussing aspects of medical terms.

This research was supported in part by the NLM Training Grant in Medical Informatics (T15 LM07041-19). Ted Pedersen's participation in this project was supported by the NSF Faculty Early Career Development Award (#0092784).

References

S. Banerjee and T. Pedersen. 2003. The design, implementation, and use of the Ngram Statistic Package. In *Proc. of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February.

A.M. Buckeridge and R.F.E. Sutcliffe. 2002. Disambiguating noun compounds with latent semantic indexing. *International Conference On Computational Linguistics*, pages 1–7.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

D.A. Evans and C. Zhai. 1996. Noun-phrase analysis in unrestricted text for information retrieval. *Proc. of the 34th conference of ACL*, pages 17–24.

D. Hindle and M. Rooth. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics*, 19(1):103–120.

M. Lapata and F. Keller. 2004. The Web as a Baseline: Evaluaing the Performance of Unsupervised Web-based Models for a Range of NLP Tasks. *Proc. of HLT-NAACL*, pages 121–128.

M. Lapata, S. McDonald, and F. Keller. 1999. Determinants of Adjective-Noun Plausibility. *Proc. of the 9th Conference of the European Chapter of ACL*, 30:36.

M. Lauer and M. Dras. 1994. A Probabilistic Model of Compound Nouns. *Proc. of the 7th Australian Joint Conference on AI*.

M. Lauer. 1995. Corpus Statistics Meet the Noun Compound: Some Empirical Results. *Proc. of the 33rd Annual Meeting of ACL*, pages 47–55.

M. Liberman and R. Sproat. 1992. The stress and structure of modified noun phrases in English. *Lexical Matters, CSLI Lecture Notes*, 24:131–181.

M.P. Marcus. 1980. *Theory of Syntactic Recognition for Natural Languages*. MIT Press Cambridge, MA, USA.

B.T. McInnes. 2004. Extending the log-likelihood ratio to improve collocation identification. Master's thesis, University of Minnesota.

R. Moore. 2004. On log-likelihood-ratios and the significance of rare events. In Dekang Lin and Dekai Wu, editors, *Proc. of EMNLP 2004*, pages 333–340, Barcelona, Spain, July. Association for Computational Linguistics.

P. Nakov and M. Hearst. 2005. Search engine statistics beyond the n-gram: Application to noun compound bracketing. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 17–24, Ann Arbor, Michigan, June. Association for Computational Linguistics.

T. Pedersen, M. Kayaalp, and R. Bruce. 1996. Significant lexical relationships. In Howard Shrobe and Ted Senator, editors, *Proc. of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference, Vol. 2*, pages 455–460, Menlo Park, California. AAAI Press.

T. Pedersen, R. Bruce, and J. Wiebe. 1997. Sequential model selection for word sense disambiguation. In *Proc. of the Fifth Conference on Applied Natural Language Processing*, pages 388–395, Washington, DC, April.

J. Pustejovsky, P. Anick, and S. Bergler. 1993. Lexical semantic techniques for corpus analysis. *Computational Linguistics*, 19(2):331–358.

A. Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Lnaguage Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

P. Resnik and M. Hearst. 1993. Structural Ambiguity and Conceptual Relations. *Proc. of the Workshop on Very Large Corpora: Academic and Industrial Perspectives, June*, 22(1993):58–64.

P.S. Resnik. 1993. *Selection and Information: A Class-Based Approach to Lexical Relationships*. Ph.D. thesis, University of Pennsylvania.

A.P. Ruggieri, S. Pakhomov, and C.G. Chute. 2004. A Corpus Driven Approach Applying the "Frame Semantic" Method for Modeling Functional Status Terminology. *Proc. of MedInfo*, 11(Pt 1):434–438.

M.Q. Stearns, C. Price, KA Spackman, and AY Wang. 2001. SNOMED clinical terms: overview of the development process and project status. *Proc AMIA Symp*, pages 662–6.

S. S. Wilks. 1938. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, March.

The Role of Roles in Classifying Annotated Biomedical Text

Son Doan, Ai Kawazoe, Nigel Collier
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
{doan,zoeai,collier}@nii.ac.jp

Abstract

This paper investigates the roles of named entities (NE's) in annotated biomedical text classification. In the annotation schema of BioCaster, a text mining system for public health protection, important concepts that reflect information about infectious diseases were conceptually analyzed with a formal ontological methodology. Concepts were classified as Types, while others were identified as being Roles. Types are specified as NE classes and Roles are integrated into NEs as attributes. We focus on the Roles of NEs by extracting and using them in different ways as features in the classifier. Experimental results show that: 1) Roles for each NE greatly helped improve performance of the system, 2) combining information about NE classes with their Roles contribute significantly to the improvement of performance. We discuss in detail the effect of each Role on the accuracy of text classification.

1 Introduction

Today, the Internet is a powerful tool for discovering novel information via news feed providers. This is becoming increasingly important for the public health domain because it can help to detect emerging and re-emerging diseases. In infectious disease surveillance systems such as the Global Public Health Intelligence Network (GPHIN) system (Public Health Agency of Canada, 2004) and ProMed-Mail (International Society for Infectious Diseases, 2001), the detection and tracking of outbreaks using the Internet has been proven to be a key source of information for public health workers, clinicians, and researchers interested in communicable diseases. The basis for such systems is the monitoring of a large number of news articles simultaneously. The classification of news articles into disease-related or none disease-related classes is the first stage in any automated approach to this task. In practice though there are a large number of news articles whose main subject is related to diseases but which should not necessarily be notified to users together with a relatively small number of high priority articles that experts should be actively alerted to. Alerting criteria broadly include news related to newly emerging diseases, the spread of diseases across international borders, the deliberate release of a human or engineered pathogen, etc. The use of only raw text in the classification process inevitably fails to resolve many subtle ambiguities, for example semantic class ambiguities in polysemous words like "virus", "fever", "outbreak", and "control" which all exhibit a variety of senses depending on context. These different senses appear with relatively high frequency in the news especially in headlines. A further challenge is that diseases can be denoted by many variant forms. Therefore we consider that the use of advanced natural language processing (NLP) techniques like named entity recognition (NER) and anaphora resolution are needed in order to achieve high classification accuracy.

Text classification is defined as the task of assigning documents into one or more predefined cat-

BioNLP 2007: Biological, translational, and clinical language processing, pages 17–24,
Prague, June 2007.

egories. As shown by (Cohen and Hersh, 2005), an accurate text classification system can be especially valuable to database curators. A document in the biomedical domain can be annotated using NER techniques with enriched semantic information in the form of NEs such as the disease, pathogen, location, and time. NER and term identification in general have been recognized as an important research topic both in the NLP and biomedical communities (Krauthammer and Nenadic, 2004). However, an investigation into the contribution of NEs on the performance of annotated biomedical text classification has remained an open question until now. There are two main reasons for this: Firstly there are a small number of open annotation schema for biomedical text, and secondly there is no benchmark annotated data for testing.

The BioCaster project (Collier, 2006) is working towards the detection and tracking of disease outbreaks from Internet news articles. Although there are several schema for biomedical text (Wilbur et al., 2006), little work has been done on developing one specifically for public health related text. BioCaster therefore provides an annotation schema that can fill this gap. Our schema, which is based on discussions with biologists, computational linguists and public health experts, helps identify entities related to infectious diseases which are then used to build up a detailed picture of events in later stages of text mining. One significant aspect of the schema is that it is based on conceptual analysis with a formal ontological methodology. As discussed in (Kawazoe et al., 2006), by applying meta-properties (Guarino and Welty, 2000a; Guarino and Welty, 2000b), our "markable" concepts are classified into "Type" and "Role". Information about Role concepts is integrated into the schema as attributes on NEs. This work takes the investigation one step forward by showing empirical evidence for the usefulness of Role concepts in a practical application.

In this paper, we focus on the task of text classification, proceeding under the simplifying assumption that given enough annotated training data for NEs and their Roles both can be automatically tagged with high accuracy. In recent years there have been many studies on text classification using general methods (Sebastiani, 2002; Yang and Liu, 1999) semi-structured texts (Kudo and Matsumoto, 2004), and XML classification (Zaki and Aggarwal, 2003). Other research has investigated the contribution of semantic information in the form of synonyms, syntax, etc. in text representation (Bloehdorn and Hotho, 2004; Hotho et al., 2003; Frürnkranz et al., 1998). Feature selection (Scott and Matwin, 1999) has also been studied. The contribution of this paper is to provide an analysis and evaluation on the Roles of NEs in annotated text classification.

The rest of this paper is organized as follows: in Section 2, we outline the BioCaster schema for the annotation of terms in biomedical text; Section 3 presents a description of the BioCaster gold standard corpus; Section 4 provides details of the method and experimental results of classification on the gold standard corpus. Finally we draw some conclusions in Section 5.

2 BioCaster Schema for Annotation of Terms in Biomedical Text

The BioCaster annotation schema is a component of the BioCaster text mining project. We have identified several important concepts that reflect information about infectious diseases, and created guidelines for annotating them as target entity classes in texts. Based on the conceptual analysis using meta-properties (rigidity, identity, and dependency) developed by Guarino and Welty (2000a; 2000b), categories of important concepts were classified as Types, i.e., properties which are rigid[1] and supply identity conditions, while others were identified as being Roles, properties which are anti-rigid[2] and dependent. The 18 categories of Type concepts are specified as NE classes which we denote here in upper case. These include PERSON, LOCATION, ORGANIZATION, TIME, DISEASE, CONDITION (status of patient such as "hospitalized" or "in stable condition"), OUTBREAK (event of group infection), VIRUS, ANATOMY (body part), PRODUCT (biological product such as "vaccine"), NONHUMAN (animals), DNA, RNA, PROTEIN, CONTROL (control measures to contain the disease), BACTERIA, CHEMICAL and SYMPTOM. The three Role concepts we explore are *case* (dis-

[1] A property is *rigid* if every instance of that property necessarily has the property, i.e. in every possible world.

[2] A property is *anti-rigid* if no instance of that property necessarily has the property.

eased person), *transmission* (source of infection) and *therapeutic* (therapeutic agent). These are integrated into the annotation schema as XML attributes which are associated with some XML elements denoting Type concepts. PERSON takes a *case* attribute, NONHUMAN and ANATOMY take *transmission*, PRODUCT takes *transmission* and *therapeutic* and CHEMICAL takes *therapeutic*. For PERSON we added another attribute *number* (number of people). Each attribute has only one value, the value of *number* is *one* or *many*, and the value of *case*, *transmission*, *therapeutic* is *true* or *false*. This is summarized in Table 1. In the rest of this paper, we call *case*, *transmission*, and *therapeutic* "Role attributes" (or "Role" for short) and *number* a "Quality attributes" (or "Quality" for short).

A NE in a biomedical text is annotated following the BioCaster annotation schema in XML format as follows,

```
<NAME cl="Named Entity"
attribute1="value1" attribute2="value2"
...  </NAME>,
```

where `"Named Entity"` is one of the names for the 18 BioCaster NEs and `attribute1, attribute2, ...` are the names of the NE's Role/Quality attributes, `"value1", "value2", ...` are values corresponding to Role/Quality attributes. Further details of the annotation guidelines are discussed in (Kawazoe et al., 2006).

3 BioCaster Gold Standard Data Corpus

The BioCaster gold standard corpus was collected from Internet news and manually annotated by two doctoral students. The annotation of a news article proceeded as follows. Firstly, NEs are annotated following the BioCaster schema and guidelines. Secondly, each annotated article is manually assigned into one of four relevancy categories: *alert*, *publish*, *check*, and *reject*. The assignment is based on guidelines that we made following discussions with epidemiologists and a survey of World Health Organization (WHO) reports (World Health Organization, 2004). These categories are currently being used operationally by the GPHIN system which is used by the WHO and other public health agencies. Where there were major differences of opinion in NE annotation or relevancy assignment between the two annotators, we consulted a public health expert in order to decide the most appropriate assignment. Finally we had a total of 500 articles that were fully annotated. While this is small compared to other data sets in text classification, we consider that it is large enough to obtain a preliminary indication about the usefulness of Role attributes.

The following is an example of an annotated article in the BioCaster gold standard corpus.

Example.

```
<DOC id="000125" language="en-us"
source="WHO" domain="health"
subdomain="disease"
date_published="2005-03-17"
relevancy="alert"> <NAME cl="DISEASE">
Acute fever </NAME> and <NAME
cl="DISEASE"> rash syndrome </NAME> in
<NAME cl="LOCATION">Nigeria</NAME> <NAME
cl="TIME"> 17 March 2005 </NAME><NAME
cl="ORGANIZATION"> WHO</NAME> has received
reports of <NAME cl="PERSON" case="true"
number="many"> 1118 cases </NAME>
including <NAME cl="PERSON" case="true"
number="many">76 deaths</NAME>case
fatality rate, 6.8% reported in 12
Local Government Areas (LGAs) of <NAME
cl="LOCATION">damawa </NAME> state, <NAME
cl="LOCATION"> Nigeria</NAME> as of <NAME
cl="TIME">28 February 2005</NAME>.  The
cases have been clinically diagnosed
as <NAME cl="DISEASE"> measles </NAME>
but no laboratory diagnosis has been
made to date.  Other states, including
<NAME cl="LOCATION">Gombe</NAME>,
<NAME cl="LOCATION">Jigawa</NAME>,<NAME
cl="LOCATION">Kaduna</NAME>, <NAME
cl="LOCATION">Kano</NAME>, and <NAME
cl="LOCATION">Kebbi</NAME> have all
reported <NAME cl="OUTBREAK"> outbreaks
</NAME> of <NAME cl="DISEASE"> measles
</NAME>...  </DOC>
```

We grouped the 500 articles into 2 categories: *reject* and *relevant*. The *reject* category corresponds simply to articles with label *reject* while the *relevant* category includes articles with labels *alert*, *publish*, and *check*. We conflated the *alert*, *publish* and *check* categories because we hypothesized that distinguishing between non-reject (relevant) categories

Named entity	Role/Quality attributes	Named entity	Role/Quality attributes
PERSON	case, number	ANATOMY	transmission
ORGANIZATION	none	SYMPTOM	none
LOCATION	none	CONTROL	none
TIME	none	CHEMICAL	therapeutic
DISEASE	none	BACTERIA	none
CONDITION	none	PRODUCT	transmission, therapeutic
NONHUMAN	transmission	DNA	none
VIRUS	none	RNA	none
OUTBREAK	none	PROTEIN	none

Table 1: Lists of Named entity classes and their Role/Quality attributes in BioCaster annotation schema.

would require higher level semantic knowledge such as pathogen infectivity and previous occurrence history which is the job of the text mining system and the end user. Finally we had a total of 269 news articles belong to the *reject* category and 231 news articles belong to the *relevant* category. The statistical information about NEs is shown in Table 2. In the table, "+" stands for the frequency of NEs in the *relevant* category and "-" stands for the frequency of NEs in the *reject* category.

4 Experiments

4.1 Method

We used the BioCaster gold standard corpus to investigate the effect of NE classes and their Role attributes on performance of classification. In order to avoid unnecessary data, we removed the first line containing DOC tag of all article in the corpus. The validation is as follows. We randomly divided the data set into 10 parts. Each of the first 9 parts has 23 articles belonging to the *relevant* category and 27 articles belonging to the *reject* category; the 10th part has 24 articles belonging to the *relevant* and 26 articles belonging to the *reject* categories. Then, we implemented 10-fold cross validation: 9 parts for training and 1 part for testing sets. For the training set we extracted NEs classes and their Roles as features to build a classifier. The remaining part was used for testing.

The classifier we use in this paper is the standard Näive Bayes classifier (Mitchell, 1997). In the preprocessing we did not use a stop list and no word stemming. The experiments were implemented in Linux OS, using the Bow toolkit (McCallum, 1996).

The details of extracting NEs and their Roles from annotated texts are the followings. For the sake of convenience, we divided features into 3 groups: Features for each NE, features for NEs with Role/Quality, and features for combined NEs with Role/Quality.

1. Features for each NE: Each NE is extracted and used with raw text as features. We denoted NE1 as features extracted from named entity NE1. For example, DISEASE1 means features are raw text and DISEASE class, VIRUS1 means features are raw text and VIRUS class. An example of features for PERSON1 is shown in Table 3.

2. Features for NEs with Role/Quality: We investigated the effect of NEs with Roles/Qualities, i.e., *case*, *number*, *therapeutic*, and *transmission*. Features are chosen as follows.

   - PERSON+case+number: Raw text and PERSON class with both Role *case* and Quality *number* are used as features.
   - PERSON+case: Raw text and PERSON class with Role *case* are used as features.
   - PERSON+number: Raw text and PERSON class and Quality *number* are used as features.
   - NONHUMAN+trans: Raw text and NONHUMAN class and Role *transmission* are used as features.
   - ANATOMY+trans: Raw text and ANATOMY class and Role *transmission* are used as features.

NE class	Frequency	Total	NE class	Frequency	Total
PERSON	+3291/-4978	8269	ANATOMY	+263/-224	487
ORGANIZATION	+1405/-3460	4865	SYMPTOM	+293/-105	398
LOCATION	+2432/-2409	4841	CONTROL	+282/-87	369
TIME	+1159/-1518	2677	CHEMICAL	+108/-185	293
DISEASE	+1164/-456	1620	BACTERIA	+136/-103	239
CONDITION	+689/-206	895	PRODUCT	+124/-74	198
NONHUMAN	+393/-344	737	DNA	+8/-55	63
VIRUS	+428/-127	555	RNA	+0/-55	55
OUTBREAK	+460/-75	535	PROTEIN	+5/-32	37

Table 2: The frequency of NE classes in the BioCaster gold standard corpus, "+" denotes the frequency in the *relevant* category and "-" denotes the frequency in the *reject* category.

Example of annotated text	`<NAME cl="ORGANIZATION"> WHO</NAME> has received reports of <NAME cl="PERSON" case="true" number="many"> 1118 cases </NAME>`
Text only	"WHO", "has", "received", "reports", "of", "1118", "cases"
PERSON1	"WHO", "has", "received", "reports", "of", "1118", "cases", "PERSON"
PERSON+case+number	"WHO", "has", "received", "reports", "of", "1118", "cases", "PERSON", "case", "number"
PERSON+case	"WHO", "has", "received", "reports", "of", "1118", "cases", "PERSON", "case"
PERSON+number	"WHO", "has", "received", "reports", "of", "1118", "cases", "PERSON", "number"

Table 3: An example of using different features for PERSON class as training data.

- PRODUCT+trans+thera: Raw text and PRODUCT class and both Roles *transmission* and *therapeutic* are used as features.
- PRODUCT+trans: Raw text and PRODUCT class and Role *transmission* are used as features.
- PRODUCT+thera: Raw text and PRODUCT class and Role *therapeutic* are used as features.
- CHEMICAL+thera: Raw text and CHEMICAL class and Role *therapeutic* are used as features.

3. Features for combined NEs with Roles. We investigate features for disease-related NEs which include DISEASE, VIRUS, BACTERIA, SYMPTOM, CONDITION, CONTROL, DNA, PROTEIN, RNA, OUTBREAK, PRODUCT, ANATOMY, NONHUMAN, CHEMICAL and features for all NEs with their Roles, i.e., *therapeutic* and *transmission*. We investigated 5 different features as follows:
   - Text only: Only raw text is used as features.
   - Text+DiseaseNEs: Raw text and all 14 NEs disease-related classes are used as features.
   - Text+DiseaseNEs+Roles: Raw text and all 14 NEs disease-related classes with Roles are used as features. We note that there are two Roles *therapeutic* and *transmission* in this case.
   - Text+AllNEs: Raw text and all NE classes are used as features.
   - Text+AllNEs+Roles: Raw text and all NE classes with Roles are used as features. In this case we have all 3 Roles *case*, *therapeutic* and *transmission*.

An example of using different features for PER-

	YES is correct	*NO* is correct
Assigned *YES*	a	b
Assigned *NO*	c	d

Table 4: A contingency table.

SON class is shown in Table 3.

4.2 Results and Discussions

The details of experimental results are shown in the following sections. We use two performance measures, standard Precision/Recall and accuracy. They are calculated based on the two-way contingency table in Table 4. In the table, a counts the assigned and correct cases, b counts the assigned and incorrect cases, c counts the not assigned but incorrect cases, and d counts the not assigned and correct cases (Yang, 1999). Then,

$$\text{Precision} = \frac{a}{a+b}, \text{ and Recall} = \frac{a}{a+c}.$$

Accuracy is defined as accuracy=$(a+d)/(a+b+c+d)$.

4.2.1 Effectiveness of Each NE Class

In order to investigate the effect of NEs on performance, we consider the baseline as the method using text only. In experiment the baseline achieved a performance of 74.40% accuracy and 64.35% Precision, 100% Recall. We can see that Recall always achieves 100% in all cases. This may be due to the small size of data. However it is interesting that we can observe the change of Precision measure - an important measure in our case. Hereafter we discuss accuracy and Precision only.

The effectiveness of each NE class is shown in Table 5. The results show that each NE does not have the same effect. Compared to the baseline, nearly half the total NEs (7/18) help improve performance while the others do not have a significant affect.

Looking at the distribution of NE frequency in Table 2, it seems that the higher the frequency of the NE class, the better the performance it provides. For example, PERSON achieved the best of all (76.80% accuracy, 66.57% Precision compared to 74.40% accuracy and 64.35% Precision when using raw text). However this trend is not always followed, for example, the TIME class tends to reduce performance when compared to raw text. This is natural as there is no obvious correlation between time and relevancy. From the result tables we can conclude that the effectiveness of each NE on the performance of classification in our corpus is decreased in the following order.

PERSON $>$ LOCATION $>$ ORGANIZATION $>$ DISEASE $>$ CONDITION = VIRUS = OUTBREAK $>$ NONHUMAN = ANATOMY = SYMPTOM = CONTROL = BACTERIA = PRODUCT = PROTEIN $>$ CHEMICAL = DNA = RNA $>$ TIME

In particular, 7 NEs, i.e., PERSON, LOCATION, ORGANIZATION, DISEASE, CONDITION, VIRUS, OUTBREAK improve performance, while TIME significantly reduces it. Two NEs DNA and RNA that have low frequency weakly reduce performance.

4.2.2 Effectiveness of Roles on Classification

In this Section we investigate the effect of each Role on performance. The experimental results are shown in Table 6. We can easily observe that Roles in NEs improved both the accuracy and Precision significantly.

We first consider the Role *case*. This Role is associated to PERSON which has highest frequency in the corpus. Role *case* helped improve the accuracy from 76.8% to 80.60%, and Precision from 66.57% to 74.43% for PERSON. This is significant when we compare to the baseline with 74.4% accuracy and 64.35% Precision. We note that PERSON has another attribute, the Quality *number*. Role *case* helps PERSON with Quality number improve the accuracy from 78.00% to 81.80% and Precision from 67.74% to 71.74%. Moreover, we can obviously draw the relative comparison about effectiveness between Role *case* and Quality *number* from these results, it yields that *case* $>$ *number*.

We proceed to investigate the effect of Roles *therapeutic* and *transmission*. Obviously we see that their effects on performance are positive. Specifically, *transmission* help NONHUMAN improve the accuracy from 74.40% to 74.60%, *therapeutic* helps CHEMICAL improve the accuracy from 74.20% to 74.40%. They both have not effects on some minor NE classes like ANATOMY and PRODUCT. If we had more training data with more of these minor NE classes we hope to see a positive effect from

Named entity	Accuracy	Pre/Rec	Named entity	Accuracy	Pre/Rec
PERSON1	76.80	66.57/100	**ANATOMY1**	74.40	64.35/100
ORGANIZATION1	75.40	65.25/100	SYMPTOM1	74.40	64.35/100
LOCATION1	75.60	65.44/100	CONTROL1	74.40	64.35/100
TIME1	73.00	63.11/100	**CHEMICAL1**	74.20	64.17/100
DISEASE1	75.00	64.89/100	BACTERIA1	74.40	64.35/100
CONDITION1	74.60	64.53/100	**PRODUCT1**	74.40	64.35/100
NONHUMAN1	74.40	64.35/100	DNA1	74.20	64.17/100
VIRUS1	74.60	64.53/100	RNA1	74.20	64.17/100
OUTBREAK1	74.60	64.53/100	PROTEIN1	74.40	64.35/100

Table 5: Performance of each NE class in which features of NEs in bold text have Role attributes.

FEATURES	Accuracy	Pre/Rec
Baseline	74.40	64.35/100
PERSON1	76.80	66.57/100
PERSON+number	78.00	67.74/100
PERSON+case	**80.60**	**74.43/100**
PERSON+case+number	**81.80**	**71.74/100**
NONHUMAN1	74.40	64.35/100
NONHUMAN+trans	**74.60**	**64.53/100**
ANATOMY1	74.40	64.35/100
ANATOMY+trans	**74.40**	**64.35/100**
PRODUCT1	74.40	64.35/100
PRODUCT+trans	**74.40**	**64.35/100**
PRODUCT+therapeutic	**74.40**	**64.35/100**
PRODUCT+trans+thera	**74.40**	**64.35/100**
CHEMICAL1	74.20	64.17/100
CHEMICAL+therapeutic	**74.40**	**64.35/100**

Table 6: Performance of Role attributes with their NEs.

Roles on them. Interestingly, while NEs associated to Roles do not improve the accuracy like NONHUMAN and CHEMICAL, their Roles helped improve the accuracy. Based on the improvements of *transmission* and *therapeutic* in Table 6, we can draw their effectiveness are the same on their NEs, that is *therapeutic* $=$ *transmission*.

When we compare the effect of all Roles on performance, we can see that the improvements of Role *case* and also Quality *number* are much higher than the improvements of Roles *therapeutic* and *transmission*. We think this is because the frequency of PERSON (NE associated to Role *case* and Quality *number*) is higher than the frequency of NEs which

FEATURES	Accuracy	Pre/Rec
Baseline	74.40	64.35/100
Text+DiseaseNEs	75.80	65.63/100
Text+DiseaseNEs+Roles	**76.20**	**66.00/100**
Text+AllNEs	79.40	69.16/100
Text+AllNEs+Roles	**84.40**	**74.76/100**

Table 7: The performance of combined NEs with their Roles.

are associated to Roles *therapeutic* and *transmission* in the corpus. Then, we can have the effect of Roles/Qualities is in the order *case* $>$ *number* $>$ *therapeutic* $=$ *transmission*.

4.2.3 Effectiveness of Combined NEs with Roles

We continue to investigate the effectiveness of Roles for combined NEs. The experimental results are given in Table 7. We note that there are two Roles *therapeutic* and *transmission* in disease-related NE classes, and all 3 Roles *case*, *therapeutic* and *transmission* in all NE classes.

We can easily see that Roles improved performance of text classification significantly. In details, for disease-related NE classes, Roles *therapeutic* and *transmission* helped to improve the accuracy from 74.40% to 76.20%, and Precision from 64.35% to $66.\%$ compared to the baseline. For all NE classes, all 3 Roles *case*, *therapeutic*, and *transmission* help to improve the accuracy from 74.40% to 84.40% and Precision from 64.35% to 74.76%. We conclude that all 3 Roles achieved the best results in performance.

5 Conclusion

This paper has focused on the contribution of Roles in biomedical annotated text classification. The experimental results indicated that:

1. Roles of each NE greatly help improve performance of the system.

2. The effect of Role/Quality attributes on classification was decreased in the order as follows: *case* $>$ *number* $>$ *therapeutic* $=$ *transmission*.

3. Combined NE classes with Roles contribute significantly to the improvement of performance.

Acknowledgments

The authors wish to thank Mika Shigematsu and Kiyosu Taniguchi at the National Institute of Infectious Diseases for useful discussions. This work was supported by Grants-in-Aid from the Japan Society for the Promotion of Science (grant no. 18049071).

References

S. Bloehdorn and A. Hotho. 2004. Boosting for text classification with semantic features. In *Proc. of the Workshop on Mining for and from the Semantic Web at the 10th ACM SIGKDD 2004*, pages 70–87.

A.M. Cohen and W.R. Hersh. 2005. A survey of current work in biomedical text mining. *Briefing in BioInformatics*, 6(3):57–71.

N. Collier. 2006. BioCaster text mining project. `http://biocaster.nii.ac.jp`.

J. Frürnkranz, T. Mitchell, and E. Riloff. 1998. A case study in using linguistic phrases for text categorization on the WWW. In *Working Notes of the AAAI/ICML Workshop on Learning for Text Categorization*, pages 5–13.

N. Guarino and C. Welty. 2000a. A formal ontology of properties. In *Proceedings of the 2000 Conference on Knowledge Engineering and Knowledge Management (EKAW-2000)*, pages 97–112.

N. Guarino and C. Welty. 2000b. Ontological analysis of taxonomic relations. In *Proceedings of the International Conference on Conceptual Modeling*, pages 210–224.

A. Hotho, S. Staab, and G. Stumme. 2003. WordNet improves text document clustering. In *Proc. of the SIGIR 2003 Semantic Web Workshop, 2003*.

International Society for Infectious Diseases. 2001. Promed mail. `http://www.promedmail.org`.

A. Kawazoe, L. Jin, M. Shigematsu, R. Barrero, K. Taniguchi, and N. Collier. 2006. The development of a schema for the annotation of terms in the BioCaster disease detection/tracking system. In *Proceedings of the International Workshop on Biomedical Ontology in Action (KR-MED 2006)*, pages 77–85.

M. Krauthammer and G. Nenadic. 2004. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6):512–526.

T. Kudo and Y. Matsumoto. 2004. A boosting algorithm for classification of semi-structured text. In *Proceedings of the 2004 Conference on Empirical Methods in NLP*, pages 301–308.

A.K. McCallum. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`.

T.M. Mitchell. 1997. *Machine Learning*. McGraw-Hill.

Public Health Agency of Canada. 2004. Global Public Heath Intelligence Network (GPHIN). `http://www.gphin.org`.

S. Scott and S. Matwin. 1999. Feature engineering for text classification. In *Proc. of International Conference on Machine Learning 1999*, pages 379–388.

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing survey*, 34(1):1–47.

W. J. Wilbur, A. Rzhetsky, and H. Shatkay. 2006. New directions in biomedical text annotation: definition, guidelines and corpus construction. *BMC Bioinformatics*, 7(356):1471–2105.

World Health Organization. 2004. ICD10, International Statistical Classification of Diseases and Related Health Problems, Tenth Revision.

Y. Yang and X. Liu. 1999. A re-examination of text categorization methods. In *Proc. of 22th ACM Int'l. Conf. on Research and Development in Information Retrieval*, pages 42–49.

Y. Yang. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval Journal*, 1:69–90.

M.J. Zaki and C.C. Aggarwal. 2003. XRules: an effective structural classifier for XML data. In *Proceedings of the ninth ACM SIGKDD International Conference, 2003*, pages 316–325.

On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA

Sampo Pyysalo, Filip Ginter, Katri Haverinen, Juho Heimonen, Tapio Salakoski
Department of Information Technology
University of Turku,
Joukahaisenkatu 3-5
20014 Turku, Finland
first.last@utu.fi

Veronika Laippala
Department of French Studies
University of Turku,
Henrikinkatu 2
20014 Turku, Finland
veronika.laippala@utu.fi

Abstract

Several incompatible syntactic annotation schemes are currently used by parsers and corpora in biomedical information extraction. The recently introduced Stanford dependency scheme has been suggested to be a suitable unifying syntax formalism. In this paper, we present a step towards such unification by creating a conversion from the Link Grammar to the Stanford scheme. Further, we create a version of the BioInfer corpus with syntactic annotation in this scheme. We present an application-oriented evaluation of the transformation and assess the suitability of the scheme and our conversion to the unification of the syntactic annotations of BioInfer and the GENIA Treebank.

We find that a highly reliable conversion is both feasible to create and practical, increasing the applicability of both the parser and the corpus to information extraction.

1 Introduction

One of the main challenges in biomedical information extraction (IE) targeting entity relationships such as protein-protein interactions arises from the complexity and variability of the natural language statements used to express such relationships. To address this complexity, many biomedical IE systems (Alphonse et al., 2004; Rinaldi et al., 2004; Fundel et al., 2007) and annotated corpora (Kim et al., 2003; Aubin, 2005; Pyysalo et al., 2007) incorporate full syntactic analysis. However, there are significant differences between the syntactic annotation schemes employed. This leads to difficulties in sharing data between corpora and establishing the relative performance of parsers as well as to a lack of interchangeability of one parser for another in IE systems, among other issues.

Syntax formalisms are broadly divided into constituency and dependency. Constituency schemes are dominant in many fields and are unified under the established Penn Treebank (PTB) scheme (Bies et al., 1995). However, dependency schemes have been suggested to be preferable in IE, as they represent the semantic structure of the sentences more directly (see, e.g., de Marneffe *et al.* (2006)). Further, Lin (1998) argues for dependency-based evaluation of both dependency and constituency parsers since it allows evaluation metrics that are more relevant to semantic interpretation as well as intuitively more meaningful. Even though there is clearly a need for a unifying scheme for dependency comparable to that of PTB for constituency, no widely adopted standard currently exists.

In this paper, we present a step towards unifying the diverse syntax schemes in use in IE systems and corpora such as the GENIA Treebank[1] and the recently introduced BioInfer corpus (Pyysalo et al., 2007). Clegg and Shepherd (2007) have recently proposed to use the Stanford dependency scheme (de Marneffe et al., 2006) as a common, application-oriented syntax representation. To assess this choice, we develop a set of conversion rules for transforming the Link Grammar (LG) dependency scheme (Sleator and Temperley, 1993) to

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/ ∼genia

BioNLP 2007: Biological, translational, and clinical language processing, pages 25–32,
Prague, June 2007.

the Stanford scheme and then create a version of the BioInfer corpus in the Stanford scheme by applying the conversion rules and manually correcting the errors. By making the BioInfer corpus available in the Stanford scheme, we also increase the value of the corpus for biomedical IE. The transformation has the further benefit of allowing Link Grammar output to be normalized into a more application-oriented form. Finally, to assess the practical value of the conversion method and of the BioInfer syntactic annotation in the Stanford scheme, we compare the Charniak-Lease constituency parser[2] (Charniak and Lease, 2005) and BioLG,[3] an adaptation of LG (Pyysalo et al., 2006), on the newly unified dataset combining the constituency-annotated GENIA Treebank with the dependency-annotated BioInfer corpus.

The transformation rules and software as well as the Stanford annotation of the BioInfer corpus, the main practical results of this work, are freely available at `http://www.it.utu.fi/BioInfer`.

2 Motivation

To support the development of IE systems, it is important for a corpus to provide three key types of annotation capturing the named entities, their relationships and the syntax. To our knowledge, there are only two corpora in the biomedical domain that currently provide these three annotation types simultaneously, BioInfer and LLL (Aubin, 2005). In addition, GENIA, the *de facto* standard domain corpus for named entity recognition and syntactic analysis, is in the process of adding a relationship annotation. The corpora have different strengths; BioInfer provides a detailed relationship annotation, while GENIA has a broader coverage of named entities and a larger treebank. Unifying the syntactic annotations of these two corpora allows these strengths to be combined.

The BioInfer syntactic annotation follows the LG dependency scheme, addressing the recent interest in LG in the biomedical NLP community (Ding et al., 2003; Alphonse et al., 2004; Aubin et al., 2005). However, the LG scheme has been criticized for being oriented more towards structural than semantic relations and having excessively detailed link types whose functional meaning and value for semantic analysis is questionable (Schneider, 1998; de Marneffe et al., 2006). Our experience with LG leads us to largely agree with these criticisms.

De Marneffe *et al.* (2006) have recently introduced a transformation from PTB to the Stanford scheme. Clegg and Shepherd (2007) have applied this transformation to perform a dependency-based comparison of several statistical constituency parsers on the GENIA Treebank and have argued for the adoption of the Stanford scheme in biomedical IE. Moreover, the IE system of Fundel *et al.* (2007), which employs the Stanford scheme, was shown to notably outperform previously applied systems on the LLL challenge dataset, finding an F-score of 72% against a previous best of 54%. This further demonstrates the suitability of the Stanford scheme to IE applications.

3 Dependency schemes

In this section, we present the Stanford and LG dependency schemes and discuss their relative strengths.

3.1 Stanford dependency scheme

A parse in the Stanford scheme (SF) is a directed graph where the nodes correspond to the words and the edges correspond to pairwise syntactic dependencies between the words. The scheme defines a hierarchy of 48 grammatical relations, or dependency types. The most generic relation, *dependent*, can be specialized as *auxiliary*, *argument*, or *modifier*, which again have several subtypes (de Marneffe et al., 2006).

The Stanford conversion transforms phrase structure parses into the Stanford scheme. First, the semantic head of each constituent is identified using head rules similar to those of Collins (1999) and untyped dependencies are then extracted and labeled with the most specific grammatical relations possible using Tregex rules (Levy and Andrew, 2006).

The system additionally provides a set of *collapsing rules*, suggested to be beneficial for IE applications (de Marneffe et al., 2006; Clegg and Shepherd, 2007). These rules collapse some dependencies by incorporating certain parts of speech (mostly

[2] `http://nlp.stanford.edu/software/`, version 1.5.1

[3] `http://www.it.utu.fi/BioLG`, version 1.2.0

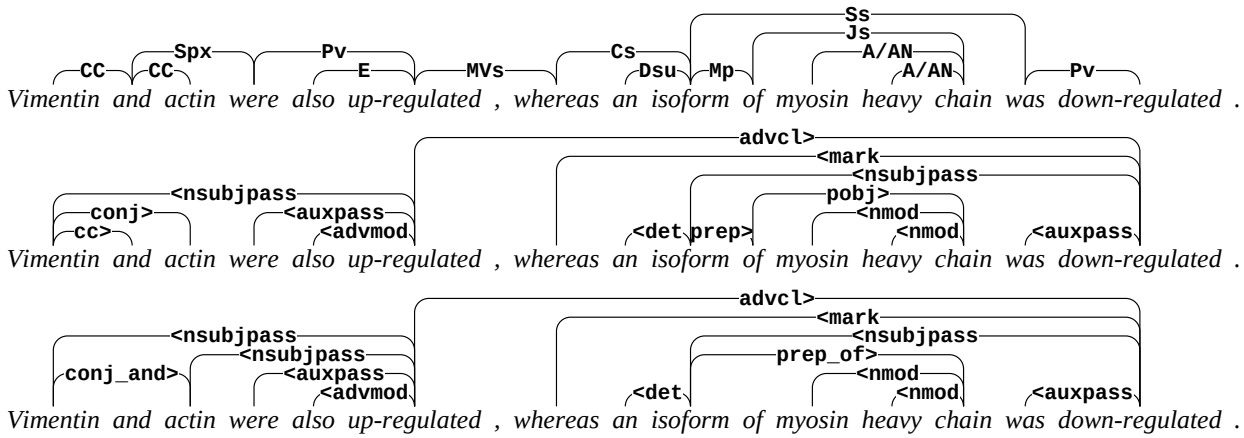


Figure 1: A sentence from the BioInfer corpus with its LG linkage (top), the Stanford parse (middle), and the collapsed Stanford parse (bottom). The $<$ and $>$ symbols denote the direction of dependencies.

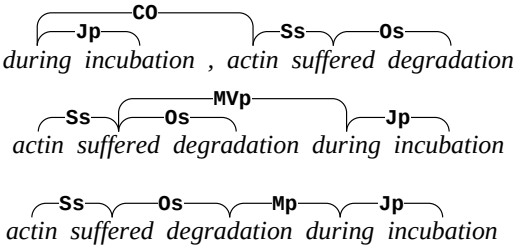


Figure 2: Variation in the link type connecting a preposition: *CO* to the main noun in topicalized prepositional phrases, *MVp* when modifying a verb, and *Mp* when modifying a noun.

conjunctions and prepositions) in grammatical relations. This is realized by combining two relations and denominating the resulting dependency with a type based on the word to which the original two relations were linked (see Figure 1).

In the LG-SF conversion, we target the uncollapsed Stanford scheme, as the collapsing rules have already been developed and reported by de Marneffe *et al.*; reimplementing the collapsing would be an unnecessary duplication of efforts. Also, the collapsed relations can be easily created based on the uncollapsed ones, whereas reversing the conversion would be more complicated.

3.2 LG dependency scheme

Link Grammar (Sleator and Temperley, 1993) is closely related to dependency formalisms. It is based on the notion of typed *links* connecting words. While links are not explicitly directional, the roles of the words can be inferred from their left-to-right order and the link type. An LG parse, termed *linkage*, consists of a set of links that connect the words so that no two links cross or connect the same two words. When discussing LG, we will use the terms dependency and link interchangeably.

Compared to the 48 dependency types of the Stanford scheme, the LG English grammar defines over 100 main link types which are further divided into 400 subtypes. The unusually high number of distinct types is one of the properties of the LG English grammar that complicate the application of LG in information extraction. Consider, for instance, the case of prepositional phrase attachment illustrated in Figure 2, where all the alternative attachment structures receive different types. Arguably, this distinction is unimportant to current IE systems and therefore should be normalized. This normalization is inherent in the Stanford scheme, where the preposition always attaches using a *prep* dependency.

In contrast to such unnecessarily detailed distinctions, in certain cases LG types fail to make semantically important distinctions. For instance, the *CO* link type is used to mark almost all clause openers, not distinguishing between, for example, adverbial and prepositional openers.

4 Our contributions

In this section, we describe the LG-SF conversion as well as SF BioInfer, the BioInfer corpus syntactic

annotation in the Stanford scheme. These are the two primary contributions of this study.

4.1 LG-SF conversion

The LG-SF conversion transforms the undirected LG links into directed dependencies that follow the Stanford scheme. The transformation is based on handwritten rules, each rule consisting of a pattern that is matched in the LG linkage and generating a single dependency in the Stanford parse. Since the conversion rules only refer to the LG linkage, they do not influence each other and are applied independently in an arbitrary order. The pattern of each rule is expressed as a set of positive or negative constraints on the presence of LG links. The constraints typically restrict the link types and may also refer to the lexical level, restricting only to links connecting certain word forms. Since LG does not define link directionality, the patterns refer to the left-to-right order of tokens and the rules must explicitly specify the directionality of the generated SF dependencies.

As an example, let us consider the rule $[X \overset{Pv}{\text{—}} Y] \Rightarrow Y \overset{auxpass}{\rightarrow} X$. The pattern matches two tokens connected with an LG link of type *Pv* and generates the corresponding directed *auxpass* dependency. This rule applies twice in the linkage in Figure 1. It is an example of a rare case of a one-to-one correspondence between an LG and an SF type. Many-to-many correspondences are much more common: in these cases, rules specify multiple restrictions and multiple rules are needed to generate all instances of a particular dependency type. As a further example, we present the three rules below, which together generate all left-to-right *prep* dependencies. An exclamation mark in front of a restriction denotes a negative restriction, i.e., the link must not exist in order for the rule to apply. The link types are specified as regular expressions.

$$[A \overset{Mp|MX[a\text{-}z]x}{\text{—}} B]![B \overset{Cs}{\text{—}} C]![A \overset{RS}{\text{—}} D] \Rightarrow A \overset{prep}{\rightarrow} B$$

$$[A \overset{OF|MVx}{\text{—}} B]![A \overset{RS}{\text{—}} C] \Rightarrow A \overset{prep}{\rightarrow} B$$

$$[A \overset{MVp}{\text{—}} B]![A \overset{RS}{\text{—}} C]![C \overset{MVl}{\text{—}} A] \Rightarrow A \overset{prep}{\rightarrow} B$$

The first of the above three rules generates the *prep* dependency in the parse in Figure 1, with *A=isoform* and *B=of*. The variables *C* and *D* are not bound to any tokens in this sentence, as they only occur in negative restrictions.

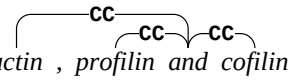


Figure 3: Example of a structure where the relative order of the first two tokens cannot be resolved by the rules.

To resolve coordination structures, it is crucial to recognize the leftmost coordinated element, i.e. the head of the coordination structure in the SF scheme. However, the conversion rule patterns are unable to capture general constraints on the relative order of the tokens. For instance, in the linkage in Figure 3, it is not possible to devise a pattern only matching one of the tokens *actin* and *profilin*, while not matching the other. Therefore, we perform a pre-processing step to resolve the coordination structures prior to the application of the conversion rules. After the pre-processing, the conversion is performed with the lp2lp software (Alphonse et al., 2004), previously used to transform LG into the LLL competition format (Aubin, 2005).

In the development of the LG-SF conversion and SF BioInfer, we make the following minor modifications to the Stanford scheme. The scheme distinguishes nominal and adjectival pre-modifiers of nouns, a distinction that is not preserved in the BioInfer corpus. Therefore, we merge the nominal and adjectival pre-modifier grammatical relations into a single relation, *nmod*. For the same reason, we do not distinguish between apposition and abbreviation, and only use the *appos* dependency type. Finally, we do not annotate punctuation.

Schneider (1998) has previously proposed a strategy for identifying the head word for each LG link, imposing directionality and thus obtaining a dependency graph. Given the idiosyncrasies of the LG linkage structures, this type of transformation into dependency would clearly not have many of the normalizing benefits of the LG-SF transformation.

4.2 SF BioInfer

For creating the BioInfer corpus syntactic annotation in the Stanford scheme, the starting point of the annotation process was the existing manual annotation of the corpus in the LG scheme to which we applied the LG-SF conversion described in Section 4.1. The resulting SF parses were then manu-

ally corrected by four annotators. In the manual correction phase, each sentence was double-annotated, that is, two annotators corrected the converted output independently. All disagreements were resolved jointly by all annotators.

To estimate the annotation quality and the stability of the SF scheme, we determined annotator agreement as precision and recall measured against the final annotation. The average annotation precision and recall were 97.5% and 97.4%, respectively. This high agreement rate suggests that the task was well-defined and the annotation scheme is stable.

The BioInfer corpus consists of 1100 sentences and, on average, the annotation consumed approximately 10 minutes per sentence in total.

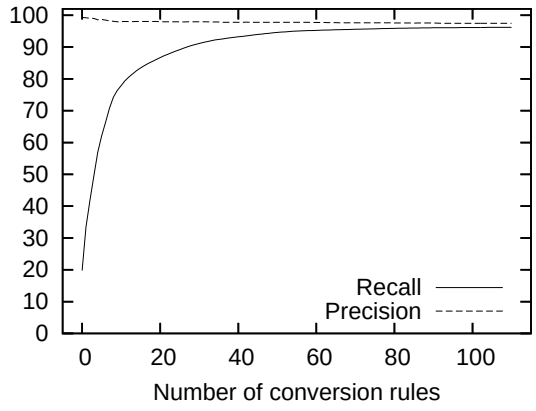


Figure 4: Cumulative precision and recall of the conversion rules.

5 Evaluation

In this section, we first evaluate the LG-SF conversion. We then present an evaluation of the Charniak-Lease constituency parser and the BioLG dependency parser on BioInfer and GENIA.

5.1 Evaluation of the conversion rules

In the evaluation of the conversion rules against the gold standard SF BioInfer annotation, we find a precision of 98.0% and a recall of 96.2%. Currently, the LG-SF conversion consists of 114 rules, each of which specifies, on average, 4.4 restrictions. Altogether the rules currently generate 32 SF dependency types, thus averaging 3.5 rules per SF type. Only 9 of the SF types are generated by a single rule, while the remaining require several rules. We estimate that the current ruleset required about 100 hours to develop.

In Figure 4, we show the cumulative precision and recall of the rules when added in the descending order of their recall. Remarkably, we find that a recall of 80% is reached with just 13 conversion rules, 90% with 28 rules, and 95% with 56 rules. These figures demonstrate that while the SF and LG schemes are substantially different, a high-recall conversion can be obtained with approximately fifty carefully crafted rules. Additionally, while precision is consistently high, the highest-recall rules also have the highest precision. This may be related to the fact that the most common SF dependency types have a straightforward correspondence in LG types.

A common source of errors in the LG-SF conversion are the Link Grammar *idiomatic expressions*, which are analyzed as a chain of *ID* links (0.7% of all links in the BioInfer corpus) and connected to the linkage always through their last word. Some examples of LG idiomatic expressions include *each other, no one, come of age, gotten rid of, for good,* and *the like*. These expressions are often problematic in the SF conversion as well. We did not attempt any wide-coverage systematic resolution of the idiomatic expressions and, apart from the most common cases such as *in vitro*, we preserve the LG structure of connecting these expressions through their last word. We note, however, that the list of idiomatic LG expressions is closed and therefore a case-by-case resolution leading to a full coverage is possible, although not necessarily practical.

Similar to the LG idiomatic expressions are the SF *dep* dependencies, generated when none of the SF rules assigns a more specific type. In most cases, *dep* is a result of a lack of coverage of the SF conversion rules typically occurring in rare or idiomatic expressions. We assume that many of the *dep* dependencies will be resolved in the future, given that the SF conversion and the SF dependency scheme itself are presented by the authors as a work in progress. Therefore, we do not attempt to replicate most of the SF *dep* dependencies with the LG-SF conversion rules; much of the effort would be obsoleted by the progress of the SF conversion. The *dep* dependencies account for 23% of the total 3.8% of dependencies not recovered by the LG-SF conversion.

	Charniak-Lease			BioLG		
corpus	Prec.	Rec.	F	Prec.	Rec.	F
GENIA	81.2	81.3	81.3	76.9	72.4	74.6
BioInfer	78.4	79.9	79.4	79.6	76.1	77.8

Table 1: Parser performance. Precision, recall and F-measure for the two parsers on the two corpora.

	BioLG		
scheme	Prec.	Rec.	F
LG	78.2	77.2	77.7
SF	79.6	76.1	77.8

Table 2: BioLG performance on the BioInfer corpus with and without the LG-SF conversion.

5.2 Evaluated parsers and corpora

The Charniak-Lease parser is a statistical constituency parser developed by Charniak and Lease (2005). It is an adaptation of the Charniak parser (Charniak, 1999) to the biomedical domain. For example, it uses a POS-tagger trained on the GENIA corpus, although the parser itself has been trained on the Penn Treebank. The Charniak-Lease parser is of particular interest, because in a recent comparison performed by Clegg and Shepherd (2007) on the GENIA Treebank, it was the best performing of several state-of-the-art statistical constituency parsers.

The LG parser is a rule-based dependency parser with a broad coverage grammar of newspaper-type English. It has no probabilistic component and does not perform pruning of ambiguous alternatives during parsing. Instead, the parser generates all parses accepted by the grammar. Simple heuristics are applied to rank the alternative parses.

Here, we evaluate a recently introduced adaptation of LG to the biomedical domain, BioLG (Pyysalo et al., 2006), incorporating the GENIA POS tagger (Tsuruoka et al., 2005) as well as a number of modifications to lexical processing and the grammar.

To facilitate the comparison of results with those of Clegg and Shepherd, we use their modified subset of GENIA Treebank.[4] As 600 of the 1100 BioInfer sentences have previously been used in the development of the BioLG parser, we only use the remaining 500 blind sentences of BioInfer in the evaluation.

5.3 Parser performance

To evaluate the performance of the parsers, we determined the *precision*, *recall* and *F-measure* by comparing the parser output against the corpus gold standard dependencies. The matching criterion required that the correct words are connected and that the direction and type of the dependency are correct. The dependency-based evaluation results for the Charniak-Lease and BioLG parsers on the GENIA and BioInfer corpora are shown in Table 1. We note that Clegg and Shepherd (2007) report 77% F-score performance of Charniak-Lease on the GENIA corpus, using the collapsed variant of the SF scheme. We replicated their experiment using the uncollapsed variant and found an F-score of 80%. Therefore, most of the approximately 4% difference compared to our finding reported in Table 1 is due to this difference in the use of collapsing, with our modifications to the SF scheme having a lesser effect. The decrease in measured performance caused by the collapsing is, however, mostly an artifact caused by merging several dependencies into one; a single mistake of the parser can have a larger effect on the performance measurement.

We find that while the performance of the Charniak-Lease parser is approximately 2 percentage units better on GENIA than on BioInfer, for BioLG we find the opposite effect, with performance approximately 3 percentage units better on BioInfer. Thus, both parsers perform better on the corpora closer to their native scheme. We estimate that this total 5 percentage unit divergence represents an upper limit to the evaluation bias introduced by the two sets of conversion rules. We discuss the possible causes for this divergence in Section 5.4.

To determine whether the differences between the two parsers on the two corpora were statistically significant, we used the Wilcoxon signed-ranks test for F-score performance using the Bonferroni correction for multiple comparisons ($N = 2$), following the recent recommendation of Demšar (2006). We find that the Charniak-Lease parser outperforms BioLG statistically significantly on both the GENIA corpus ($p \ll 0.01$) and on the BioInfer corpus

[4] `http://chomsky-ext.cryst.bbk.ac.uk/andrew/downloads.html`

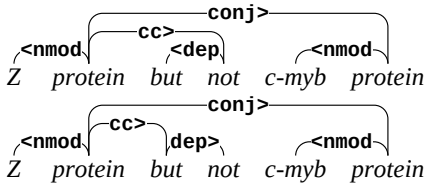


Figure 5: Example of divergence on the interpretation of the Stanford scheme. Above: GENIA and Stanford conversion interpretation. Below: BioInfer and LG-SF rules interpretation.

($p < 0.01$). Thus, the relative performance of the parsers can, in this case, be established even in the presence of opposing conversion biases on the two corpora.

In Table 2, we present an evaluation of the BioLG parser with and without the LG-SF conversion, specifically evaluating the effect of the conversion presented in this study. Here we find a substantially more stable performance, including even an increase in precision. This further validates the quality of the conversion rules.

Finally, we note that the processing time required to perform the conversions is insignificant compared to the time consumed by the parsers.

5.4 Discussion

Evaluating BioLG on GENIA and the Charniak-Lease parser on BioInfer includes multiple sources of divergence. In addition to parser errors, differences can be created by the LG-SF conversion and the Stanford conversion. Moreover, in examining the outputs we identified that a further source of divergence is due to differing interpretations of the Stanford scheme. One such difference is illustrated in Figure 5. Here the BioLG parser with the LG-SF conversion produces an analysis that differs from the result of converting the GENIA Treebank analysis by the Stanford conversion. This is due to the Stanford conversion producing an apparently flawed analysis that is not replicated by the LG-SF conversion. In certain cases of this type, the lack of a detailed definition of the SF scheme prevents from distinguishing between conversion errors and intentional analyses. This will necessarily lead to differing interpretations, complicating precise evaluation.

6 Conclusions

We have presented a step towards unifying syntactic annotations under the Stanford dependency scheme and assessed the feasibility of this unification by developing and evaluating a conversion from Link Grammar to the Stanford scheme. We find that a highly reliable transformation can be created, giving a precision and recall of 98.0% and 96.2%, respectively, when compared against our manually annotated gold standard version of the BioInfer corpus. We also find that the performance of the BioLG parser is not adversely affected by the conversion. Given the clear benefits that the Stanford scheme has for domain analysis, the conversion increases the overall suitability of the parser to IE applications. Based on these results, we conclude that converting to the Stanford scheme is both feasible and practical.

Further, we have developed a version of the BioInfer corpus annotated with the Stanford scheme, thereby increasing the usability of the corpus. We applied the LG-SF conversion to the original LG BioInfer annotation and manually corrected the errors. The high annotator agreement of above 97% precision and recall confirms the stability of the SF scheme.

We have also demonstrated that the unification permits direct parser comparison that was previously impossible. However, we found that there is a certain accumulation of errors caused by the conversion, particularly in a case when two distinct rule sets are applied. In our case, we estimate this error to be on the order of several percentage units, nevertheless, we were able to establish the relative performance of the parses with a strong statistical significance. These results demonstrate the utility of the Stanford scheme as a unifying representation of syntax. We note that an authoritative definition of the Stanford scheme would further increase its value.

Acknowledgments

We would like to thank Erick Alphonse, Sophie Aubin and Adeline Nazarenko for providing us with the lp2lp software and the LLL conversion rules. We would also like to thank Andrew Brian Clegg and Adrian Shepherd for making available the data and evaluation tools used in their parser evaluation. This work was supported by the Academy of Finland.

References

Erick Alphonse, Sophie Aubin, Philippe Bessières, Gilles Bisson, Thierry Hamon, Sandrine Laguarigue, Adeline Nazarenko, Alain-Pierre Manine, Claire Nédellec, Mohamed Ould Abdel Vetah, Thierry Poibeau, and Davy Weissenbacher. 2004. Event-Based Information Extraction for the biomedical domain: the Caderige project. In N. Collier, P. Ruch, and A. Nazarenko, editors, *COLING NLPBA/BioNLP Workshop*, pages 43–49, Geneva, Switzerland.

Sophie Aubin, Adeline Nazarenko, and Claire Nédellec. 2005. Adapting a general parser to a sublanguage. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, and N. Nikolov, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 05), Borovets, Bulgaria*, pages 89–93. Incoma, Bulgaria.

Sophie Aubin. 2005. LLL challenge - syntactic analysis guidelines. Technical report, LIPN, Université Paris Nord, Villetaneuse.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for treebank ii style. Technical report, Penn Treebank Project, University of Pennsylvania.

Eugene Charniak and Matthew Lease. 2005. Parsing biomedical literature. In R. Dale, K. F. Wong, J. Su, and O. Y. Kwong, editors, *Proceedings of the Second International Joint Conference on Natural Langage Processing, Jeju Island, Korea*, pages 58–69.

Eugene Charniak. 1999. A maximum-entropy-inspired parser. Technical report, Brown University.

Andrew Brian Clegg and Adrian Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1):24.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, and D. Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449–454.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Jing Ding, Daniel Berleant, Jun Xu, and Andy W. Fulmer. 2003. Extracting biochemical interactions from medline using a link grammar parser. In B. Werner, editor, *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 467–471. IEEE Computer Society, Los Alamitos, CA.

Katrin Fundel, Robert Kuffner, and Ralf Zimmer. 2007. RelEx–Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus–a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–182.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In N. Calzolari, K. Choukri, A. Gangemi, B. Maegaard, J. Mariani, J. Odijk, and D. Tapias, editors, *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 2231–2234.

Dekang Lin. 1998. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4(2):97–114.

Sampo Pyysalo, Tapio Salakoski, Sophie Aubin, and Adeline Nazarenko. 2006. Lexical adaptation of link grammar to the biomedical sublanguage: a comparative evaluation of three approaches. *BMC Bioinformatics*, 7(Suppl 3).

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, James Dowdall, Andreas Persidis, and Ourania Konstanti. 2004. Mining relations in the genia corpus. In *Proceedings of the Workshop W9 on Data Mining and Text Mining for Bioinformatics (ECML/PKDD'04)*, pages 61–68, Pisa, Italy.

Gerold Schneider. 1998. A linguistic comparison of constituency, dependency and link grammar. Master's thesis, University of Zürich.

Daniel D. Sleator and Davy Temperley. 1993. Parsing English with a Link Grammar. In *Third International Workshop on Parsing Technologies*.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun'ichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In P. Bozanis and E. N. Houstis, editors, *10th Panhellenic Conference on Informatics*, volume 3746, pages 382–392.

An Unsupervised Method for Extracting Domain-specific Affixes in Biological Literature

Haibin Liu Christian Blouin Vlado Kešelj

Faculty of Computer Science, Dalhousie University, Canada, {haibin,cblouin,vlado}@cs.dal.ca

Abstract

We propose an unsupervised method to automatically extract domain-specific prefixes and suffixes from biological corpora based on the use of PATRICIA tree. The method is evaluated by integrating the extracted affixes into an existing learning-based biological term annotation system. The system based on our method achieves comparable experimental results to the original system in locating biological terms and exact term matching annotation. However, our method improves the system efficiency by significantly reducing the feature set size. Additionally, the method achieves a better performance with a small training data set. Since the affix extraction process is unsupervised, it is assumed that the method can be generalized to extract domain-specific affixes from other domains, thus assisting in domain-specific concept recognition.

1 Introduction

Biological term annotation is a preparatory step in information retrieval in biological science. A biological term is generally defined as any technical term related to the biological domain. Considering term structure, there are two types of biological terms: single word terms and multi-word terms. Many systems (Fukuda et al., 1998; Franzn et al., 2002) have been proposed to annotate biological terms based on different methodologies in which determining term boundaries is usually the first task. It has been demonstrated (Jiampojamarn et al., 2005a), however, that accurately locating term boundaries is difficult. This is so because of the ambiguity of terms, and the peculiarity of the language used in biological literature.

(Jiampojamarn et al., 2005b) proposed an automatic biological term annotation system (ABTA) which applies supervised learning methods to annotate biological terms in the biological literature. Given unstructured texts in biological research, the annotation system first locates biological terms based on five word position classes, "Start", "Middle", "End", "Single" and "Non-relevant". Therefore, multi-word biological terms should be in a consistent sequence of classes "Start (Middle)* End" while single word terms will be indicated by the class "Single". Word n-grams (Cavnar and Trenkle, 1994) are used to define each input sentence into classification instances. For each element in an n-gram, the system extracts feature attributes as input for creating the classification model. The extracted feature attributes include word feature patterns(e.g., Greek letters, uppercase letters, digits and other symbols), part-of-speech (POS) tag information, prefix and suffix characters. Without using other specific domain resources, the system achieves comparable results to some other state-of-the-art systems (Finkel et al., 2004; Settles, 2004) which resort to external knowledge, such as protein dictionaries. It has been demonstrated (Jiampojamarn et al., 2005b) that the part-of-speech tag information is the most effective attribute in aiding the system to annotate biological terms because most biological terms are partial noun phrases.

The ABTA system learns the affix feature by recording only the first and the last n characters (e.g., $n = 3$) of each word in classification instances, and the authors claimed that the n characters could provide enough affix information for the term annotation task. Instead of using a certain number of characters to provide affix information, however, it is more likely that a specific list of typically used prefixes and suffixes of biological words would provide more accurate information to classifying some biological terms and boundaries. We hypothesize that

BioNLP 2007: Biological, translational, and clinical language processing, pages 33–40,
Prague, June 2007.

a more flexible affix definition will improve the performance of the taks of biological term annotation.

Inspired by (Jiampojamarn et al., 2005b), we propose a method to automatically extract domain-specific prefixes and suffixes from biological corpora. We evaluate the effectiveness of the extracted affixes by integrating them into the parametrization of an existing biological term annotation system, ABTA (Jiampojamarn et al., 2005b), to evaluate the impact on performance of term annotation. The proposed method is completely unsupervised. For this reason, we suggest that our method can be generalized for extracting domain-specific affixes from many domains.

The rest of the paper is organized as follows: In section 2, we review recent research advances in biological term annotation. Section 3 describes the methodology proposed for affix extraction in detail. The experiment results are presented and evaluated in section 4. Finally, section 5 summarizes the paper and introduces future work.

2 Related Work

Biological term annotation denotes a set of procedures that are used to systematically recognize pertinent terms in biological literature, that is, to differentiate between biological terms and non-biological terms and to highlight lexical units that are related to relevant biology concepts (Nenadic and Ananiadou, 2006).

Recognizing biological entities from texts allows for text mining to capture their underlying meaning and further extraction of semantic relationships and other useful information. Because of the importance and complexity of the problem, biological term annotation has attracted intensive research and there is a large number of published work on this topic (Cohen and Hersh, 2005; Franzn et al., 2003).

Current approaches in biological term annotation can be generalized into three main categories: lexicon-based, rule-based and learning-based (Cohen and Hersh, 2005). Lexicon-based approaches use existing terminological resources, such as dictionaries or databases, in order to locate term occurrences in texts. Given the pace of biology research, however, it is not realistic to assume that a dictionary can be maintained up-to-date. A drawback of lexicon-based approaches is thus that they are not able to annotate recently coined biological terms. Rule-based approaches attempt to recover terms by developing rules that describe associated term formation patterns. However, rules are often time-consuming to develop while specific rules are difficult to adjust to other types of terms. Thus, rule-based approaches are considered to lack scalability and generalization.

Systems developed based on learning-based approaches use training data to learn features useful for biological term annotation. Compared to the other two methods, learning-based approaches are theoretically more capable to identify unseen or multi-word terms, and even terms with various writing styles by different authors. However, a main challenge for learning-based approaches is to select a set of discriminating feature attributes that can be used for accurate annotation of biological terms. The features generally fall into four classes: (1) simple deterministic features which capture use of uppercase letters and digits, and other formation patterns of words, (2) morphological features such as prefix and suffix, (3) part-of-speech features that provide word syntactic information, and (4) semantic trigger features which capture the evidence by collecting the semantic information of key words, for instances, head nouns or special verbs.

As introduced earlier, the learning-based biological term annotation system ABTA obtained an 0.705 F-score in exact term matching on Genia corpus (v3.02)[1] which contains 2,000 abstracts of biological literature. In fact, the morphological features in ABTA are learned by recording only the first and the last n characters of each word in classification instances. This potentially leads to inaccurate affix information for the term annotation task.

(Shen et al., 2003) explored an adaptation of a general Hidden Markov Model-based term recognizer to biological domain. They experimented with POS tags, prefix and suffix information and noun heads as features and reported an 0.661 F-score in overall term annotation on Genia corpus. 100 most frequent prefixes and suffixes are extracted as candidates, and evaluated based on difference in likelihood of part of a biological term versus not. Their method results in a modest positive improvement in recognizing biological terms. Two limitations of this method are: (1) use of only a biological corpus, so

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

that the general domain-independent affixes are not removed, and (2) a supervised process of choosing a score threshold that is used in affix selection.

(Lee et al., 2003) used prefix and suffix features coupled with a dictionary-based refinement of boundaries of the selected candidates in their experiments for term annotation. They extracted affix features in a similar way with (Shen et al., 2003). They also reported that affix features made a positive effect on improving term annotation accuracy.

In this project, we consider the quality of domain-specific affix features extracted via an unsupervised method. Successful demonstration of the quality of this extraction method implies that domain-specific affixes can be identified for arbitrary corpora without the need to manually generate training sets.

3 PATRICIA-Tree-based Affix Extraction

3.1 PATRICIA Tree

The method we propose to extract affixes from biological words is based on the use of PATRICIA tree. "PATRICIA" stands for "Practical Algorithm To Retrieve Information Coded In Alphanumeric". It was first proposed by (Morrison, 1968) as an algorithm to provide a flexible means of storing, indexing, and retrieving information in a large file. PATRICIA tree uses path compression by grouping common sequences into nodes. This structure provides an efficient way of storing values while maintaining the lookup time for a key of O(*N*) in the worst case, where *N* is the length of the longest key. Meanwhile, PATRICIA tree has little restriction on the format of text and keys. Also it does not require rearrangement of text or index when new material is added. Because of its outstanding flexibility and efficiency, PATRICIA tree has been applied to many large information retrieval problems (Morrison, 1968).

In our project, all biological words are inserted and stored in a PATRICIA tree, using which we can efficiently look up specific biological word or extract biological words that share specified affixes and calculated required statistics.

3.2 Experiment Design

In this work, we have designed the experiments to extract domain-specific prefixes and suffixes of biological words from a biological corpus, and investigate whether the extracted affix information could facilitate better biological term annotation. The overall design of our experiments consists of three major processes: affix extraction, affix refining and evaluation of experimental results. It is seen that every node in PATRICIA tree contains exactly one string of 1 or more characters, which is the preceding substring of its descendant nodes. Meanwhile, every word is a path of substrings from the root node to a leaf. Therefore, we propose that every substring that can be formed from traversing the internal nodes of the tree is a potential affix.

In the affix extraction process, we first populate a PATRICIA tree using all words in the combined corpus*(CC)* of a Biological Corpus *(BC)* and a General English Corpus *(GEC)*. *GEC* is used against *BC* in order to extract more accurate biological affix information. Two PATRICIA trees are populated separately for extracting prefixes and suffixes. The suffix tree is based on strings derived by reversing all the input words from the combined corpus. All the potential prefixes and suffixes are then extracted from the populated PATRICIA trees.

In the affix refining process, for each extracted potential affix, we compute its joint probability of being both an English affix and a biological affix, $P(D = \textit{Biology}, A = \textit{Yes}|PA)$, where *D* stands for *Domain*, *A* stands for *Affix* and *PA* represents *Potential Affix*. This joint probability can be further decomposed as shown in Eq.(1). In the formula, $P\,(A = \textit{Yes}|PA)$ denotes the probability that a given potential affix is a true English affix while $P(D = \textit{Biology}|A = \textit{Yes}, PA)$ refers to the probability that a given English affix is actually a biological affix.

$$P(D = \textit{Biology}, A = \textit{Yes}|PA) = P(D{=}\textit{Biology}|A{=}\textit{Yes}, PA) \times P(A{=}\textit{Yes}|PA) \quad (1)$$

To calculate $P\,(A = \textit{Yes}|PA)$, the probabilities of prefixes and suffixes are measured separately. In linguistics, a prefix is described as a type of affix that precedes the morphemes to which it can attach (Soanes and Stevenson, 2004). Simply speaking, a prefix is a substring that can be found at the beginning of a word. Our functional definition of a prefix is a substring which precedes words existing in the English language. This can be done by enumerating, for each node, all descendant substring and assessing their existence as stand-alone words. For example, "radioimmunoassay", "radioiodine" and "radio-

labeled" are three words and have a common starting string "radio". If we take out the remaining part of each word, three new strings are obtained, "immunoassay", "iodine" and "labeled". Since all the input words are already stored in PATRICIA tree, we lookup these three strings in PATRICIA tree and find that "immunoassay", "iodine" and "labeled" are also meaningful words in the tree. This indicates that "radio" is a prefix among the input words. On the other hand, it is obvious that "radioimmunoassay" and "radioiodine" share another string "radioi". However, "mmunoassay" and "odine" are not meaningful words due to their absence in the PATRICIA tree. This suggests that "radioi" is not a prefix.

For each extracted potential prefix, $P(A = \textit{Yes}|PA)$ is computed as the proportion of strings formed by traversing all descendant nodes that are meaningful terms. In our experiments, the measure of determining a string meaningful is to look up whether the string is an existing word present in the built prefix PATRICIA tree. Algorithm 1 shows the procedure of populating a PATRICIA tree and calculating $P(A = \textit{Yes}|PA)$ for each potential prefix.

Algorithm 1 $P(A = \textit{Yes}|PA)$ for Prefix

Input: words $(w) \in$ Combined Corpus (CC)
Output: $P(A = \textit{Yes}|PA)$ for each potential prefix

```
PT = ∅                                   //PT : Patricia Trie
for all words w ∈ CC do
    PT ← Insert(w)        //Populating Patricia Trie
for all nodes n_i ∈ PT do
    PA ← String(n_i)          //Concatenate strings
                              // in nodes from root to n_i,
                              // which is a potential prefix
    T_PA ← PrefixSearch(PA)
    //T_PA : all words w ∈ CC beginning with PA
    score ← 0
    for all words w ∈ T_PA do
        if Extrstr(PA, w) in PT then
            //Extrstr() returns the remaining string
            // of w without PA
            score ++
    P(A = Yes|PA) ← score/|T_PA|
    //|T_PA| is the number of words in T_PA
```

Likewise, in linguistics a suffix is an affix that follows the morphemes to which it can attach (Soanes and Stevenson, 2004). Simply speaking, a suffix of a word is a substring exactly matching the last part of the word. Similar to the idea of calculating $P(A = \textit{Yes}|PA)$ for potential prefix, we conjecture that the extracted potential suffix could be a reasonable English suffix if the inverted strings formed from traversing the descendant nodes of the potential suffix in the suffix PATRICIA tree are meaningful words. For instance, "Calcium-dependent", "Erythropoietin-dependent" and "Ligand-dependent" share a common ending string "-dependent". Since the remaining strings of each word, "Calcium", "Erythropoietin" and "Ligand" can be found in the "forward" PATRICIA tree, "-dependent" is a potentially useful suffix.

However, it is often observable that some English words do not begin with another meaningful word but a typical prefix, for example, "alpha-bound" and "pro-glutathione". It is known that "-bound"and "-glutathione" are good suffixes in biology. "alpha" and "pro", however, are not meaningful words but typical prefixes, and in fact have been extracted when calculating $P(A = \textit{Yes}|PA)$ for potential prefix. Therefore, in order to detect and capture such potential suffixes, we further assume that if a word begins with a recognized prefix instead of another meaningful word, the remaining part of the word still has the potential to be an informative suffix. Therefore, strings "-bound" and "-glutathione" can be successfully extracted as potential suffixes. In our experiments, an extracted potential prefix is considered a recognized prefix if its $P(A = \textit{Yes}|PA)$ is greater than 0.5.

To calculate $P(D = \textit{Biology}|A = \textit{Yes}, PA)$, it is necessary to first determine true English affixes from extracted potential affixes. In our experiments, we consider that an extracted potential prefix or suffix is a recognized affix only if its $P(A = \textit{Yes}|PA)$ is greater than 0.5. It is also necessary to consider the biological corpus *BC* and the general English corpus *GEC* separately. It is assumed that a biology related affix tends to occur more frequently in words of *BC* than *GEC*. Eq.(2) is used to estimate $P(D = \textit{Biology}|A = \textit{Yes}, PA)$.

$$P(D = \textit{Biology}|A = \textit{Yes}, PA) = (\#Words\ with\ PA\ in\ BC/Size(BC))/(\#Words\ with\ PA\ in\ BC/Size(BC) + \#Words\ with\ PA\ in\ GEC/Size(GEC)), \quad (2)$$

where only *PA* with $P(A = Yes|PA)$ greater than 0.5 are used, and the number of words with a certain *PA* is further normalized by the size of each corpus.

Finally, the joint probability of each potential affix, $P(D = Biology, A = Yes|PA)$, can be used to parametrize a word beginning or ending with *PA*.

In the evaluation process of our experiments, the prefix-suffix pair with maximum joint probability values is used to parametrize a word. Therefore, each word in *BC* has exactly two values as affix feature: a joint probability value for its potential prefix and a joint probability value for its potential suffix. We then replace the original affix feature of ABTA system with our obtained joint probability values, and investigate whether these new affix information leads to equivalent or better term annotation on *BC*.

4 Results and Evaluation

4.1 Dataset and Environment

For our experiments, it is necessary to use a corpus that includes widely used biological terms and common English words. This dataset, therefore, will allow us to accurately extract the information of biology related affixes. As a proof-of-concept prototype, our experiments are conducted on two widely used corpora: Genia corpus (v3.02) and Brown corpus[2].The Genia version 3.02 corpus is used as the biological corpus *BC* in our experiments. It contains 2,000 biological research paper abstracts. They were selected from the search results in the MEDLINE database[3], and each biological term has been annotated into different terminal classes based on the opinions of experts in biology. Used as the general English corpus *GEC*, Brown corpus includes 500 samples of common English words, totalling about a million words drawn from 15 different text categories.

All the experiments were executed on a Sun Solaris server Sun-Fire-880. Our experiments were mainly implemented using Perl and Python.

4.2 Experimental Results

We extracted 15,718 potential prefixes and 21,282 potential suffixes from the combined corpus of Genia and Brown. Among them, there are 2,306 potential prefixes and 1,913 potential suffixes with joint probability value $P(D = Biology, A = Yes|PA)$ greater than 0.5. Table 1 shows a few examples of extracted potential affixes whose joint probability value is equal to 1.0. It is seen that most of these potential affixes are understandable biological affixes which directly carry specific semantic meanings about certain biological terms. However, some substrings are also captured as potential affixes although they may not be recognized as "affixes" in linguistics, for example "adenomyo" in prefixes, and "mopoiesis" in suffixes. In Genia corpus, "adenomyo" is the common beginning substring of biological terms "adenomyoma", "adenomyosis" and "adenomyotic" , while "plasias" is the common ending substring of biological terms "neoplasias" and "hyperplasias". The whole list of extracted potential affixes is available upon request.

In order to investigate whether the extracted affixes improves the performance of biological term annotation, it is necessary to obtain the experimental results of both original ABTA system and the ABTA system using our extracted affix information. In ABTA, the extraction of feature attributes is performed on the whole 2000 abstracts of Genia corpus, and then 1800 abstracts are used as training set while the rest 200 abstracts are used as testing set. The evaluation measures are precision, recall and F-score. C4.5 decision tree classifier (Alpaydin, 2004) is reported as the most efficient classifier which leads to the best performance among all the classifiers experimented in (Jiampojamarn et al., 2005b). Therefore, C4.5 is used as the main classifier in our experiments. The experimental results of ABTA system with 10 fold cross-validation based on different combinations of the original features are presented in Table 2 in which feature *"WFP"* is short for Word Feature Patterns, feature *"AC"* denotes Affix Characters, and feature *"POS"* refers to POS tag information. The setting of parameters in the experiments with ABTA is: the word n-gram size is 3, the number of word feature patterns is 3, and the number of affix characters is 4. We have reported the F-score and the classification accuracy of the experiments in the table. It is seen that there is a tendency with the experimental performance that for a multi-word biological term, the middle position is most difficult to detect while the ending position is generally easier to be identified than the starting position. The assumed reason for this tendency is that for multi-

[2]http://clwww.essex.ac.uk/w3c/corpus_ling/

[3]http://www.ncbi.nlm.nih.gov/PubMed/

Potential Prefixes				Potential Suffixes			
13-acetate	adenomyo	3-kinase	platelet	-T-cell	-alpha-activated	cytoid	-methyl
B-cell	Rel/NF-kappaB	CD28	pharmaco	-coated	mopoiesis	-bearing	lyse
endotoxin	anti-CD28	HSV-1	adenovirus	-expressed	-nonresponsive	-kappaB-mediated	-receptor
I-kappaB	VitD3	ligand	chromatin	-inducer	coagulant	-globin-encoding	glycemia
macrophage	cytokine	N-alpha-tosyl-L	hemoglobin	plasias	-soluble	-immortalized	racrine

Table 1: Examples of Extracted Potential Affixes with Joint Probability Value 1.0

word biological terms, many middle words of are seemingly unrelated to biology domain while many ending words directly indicate their identity, for instances, "receptor", "virus" or "expression".

Table 3 shows the experimental results of ABTA system after replacing the original affix feature with our obtained joint probability values for each word in Genia corpus. *"JPV"* is used to denote Joint Probability Values. It is seen that based on all three features the system achieves a classification accuracy of 87.5%, which is comparable to the results of the original ABTA system. However, the size of the feature set of the system is significantly reduced, and the classification accuracy of 87.5% is achieved based on only 18 parameters, which is 1/2 of the size of the original feature set. Meanwhle, the execution time of the experiments generally reduces to nearly half of the original ABTA system (e.g., reduces from 4 hours to 1.7 hours). Furthermore, when the feature set contains only our extracted affix information, the system reaches a classification accuracy of 81.46% based on only 6 parameters. It is comparable with the classification accuracy achieved by using only POS information in the system. In addition, Table 3 also presents the experimental results when our extracted affix information is used as an addtional feature to the original feature set. It is expected that the system performance is further improved when the four features are applied together. However, the size of the feature set increases to 42 parameters, which increases the data redundancy. This proves that the extracted affix information has a positive impact on locating biological terms, and it could be a good replacement of the original affix feature.

Moreover, we also evaluated the performance of the exact matching biological term annotation based on the obtained experimental results of ABTA system. The exact matching annotation in ABTA system is to accurately identify every biological term, including both multi-word terms and single word terms, therefore, all the word position classes of a term have to be classified correctly at the same time. An error occurring in any one of "Start" "Middle" and "End" classes leads the system to annotate multi-word terms incorrectly. Consequently, the accumulated errors will influence the exact matching annotation performance. Table 4 presents the exact matching annotation results of different combination of features based on 10 fold cross-validation over Genia corpus. It is seen that after replacing the original affix feature of ABTA system with our obtained joint probability values for each word in Genia corpus, the system achieves an 0.664 F-score on exact matching of biological term annotation, comparable to the exact matching performance of the original ABTA system. In addition, when the feature set contains only our extracted affix information, the system reaches an 0.536 F-score on exact matching. Although it is a little lower than the exact matching performance achieved by using only the original affix features in the system, the feature set size of the system is significantly reduced from 24 to 6.

In order to further compare our method with the original ABTA system, we attempted eleven different sizes of training data set to run the experiments separately based on our method and the original ABTA system. They can then be evaluated in terms of their performance on each training set size. These eleven different training set sizes are: 0.25%, 0.5%, 1%, 2.5%, 5%, 7.5%, 10%, 25%, 50%, 75% and 90%. For instance, 0.25% denotes that the training data set is 0.25% of Genia corpus while the rest 99.75% becomes the testing data set for experiments. It is observed that there are about 21 paper abstracts in training set when its size is 1% , and 52 abstracts when its size is 2.5%. It is expected that larger training set size leads to better classification accuracy of experiments.

For each training set size, we randomly extracted 10 different training sets from Genia corpus to run the experiments. We then computed the *mean classification accuracy (MCA)* of 10 obtained classification accuracies. Figure 1 was drawn to illustrate the distribution of MCA of each training set size

Feature sets	F-Measure					Classification Accuracy (%)	# Parameters
	Start	Middle	End	Single	Non		
WFP	0.467	0.279	0.495	0.491	0.864	74.59	9
AC	0.709	0.663	0.758	0.719	0.932	85.67	24
POS	0.69	0.702	0.775	0.67	0.908	83.96	3
WFP+AC	0.717	0.674	0.762	0.730	0.933	86.02	33
WFP+POS	0.726	0.721	0.793	0.716	0.923	85.96	12
AC+POS	0.755	0.741	0.809	0.732	0.930	87.14	27
WFP+AC+POS	0.764	0.745	0.811	0.749	0.933	**87.59**	**36**

Table 2: Experimental Results of Original ABTA System

Feature sets	F-Measure					Classification Accuracy (%)	# Parameters
	Start	Middle	End	Single	Non		
JPV	0.652	0.605	0.713	0.602	0.898	**81.46**	**6**
WFP+JPV	0.708	0.680	0.756	0.699	0.919	84.84	15
JPV+POS	0.753	0.740	0.805	0.722	0.928	86.92	9
WFP+JPV+POS	0.758	0.749	0.809	0.74	0.933	**87.50**	**18**
WFP+AC+POS+JPV	0.767	0.746	0.816	0.751	0.934	87.77	42

Table 3: Experimental Results of ABTA System with Extracted Affix Information

for both methods, with the incremental proportion of training data. It is noted in Figure 1 that the change patterns of MCA obtained by our method and the original ABTA system are similar. It is also seen that our method achieves marginally better classification performance when the proportion of training dat

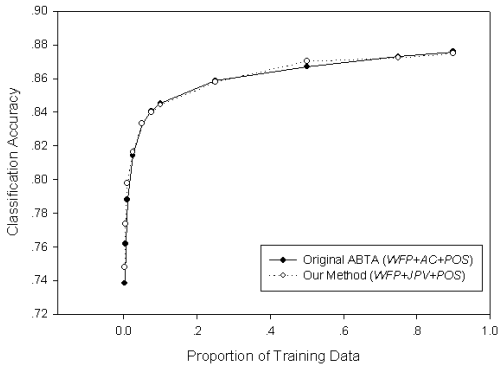


Figure 1: MCA Distribution

In order to determine if the classification performance difference between our method and the original ABTA system is statistically significant, we performed one-tailed t-Test (Alpaydin, 2004) on the classification results with our hypothesis that MCA of our proposed method is higher than MCA of original ABTA system. The significance level α is set to be the conventional value 0.05. As a result, the classification performance difference between two methods is statistically significant when the proportion of training data is 0.25%, 0.5%, 1% or 2.5%. Table 5 shows the *P* values of t-Test results for the various training set sizes. This demonstrates that the ABTA system adopting our method outperforms the original ABTA system in classification accuracy when the proportion of training data is lower than 2.5% of Genia corpus, and achieves comparable classification performance with the original ABTA system when the proportion continuously increases.

One-tailed t-Test	Training set size			
	0.25%	0.5%	1%	2.5%
P value	0.0298	0.0006	0.0002	0.0229

Table 5: One-tailed t-Test Results

5 Conclusions

In this paper, we have presented an unsupervised method to extract domain-specific prefixes and suffixes from the biological corpus based on the use of PATRICIA tree. The ABTA system (Jiampojamarn et al., 2005b) adopting our method achieves an overall classification accuracy of 87.5% in locating biological terms, and derives an 0.664 F-score in exact term matching annotation, which are all comparable to the experimental results obtained by the original ABTA system. However, our method helps the system significantly reduce the size of feature set and thus improves the system efficiency. The system also obtains a classification accuracy of 81.46% based only on our extracted affix information. This

Feature sets	Exact Matching Annotation			# Parameters
	Precision	Recall	F-score	
AC	0.548	0.571	0.559	24
WFP+AC+POS	0.661	0.673	0.667	36
JPV	0.527	0.545	0.536	6
WFP+JPV+POS	0.658	0.669	0.664	18

Table 4: Exact Matching Annotation Performance

demonstates that the affix information acheived by the proposed method is important to accurately locating biological terms.

We further explored the reliability of our method by gradually increasing the proportion of training data from 0.25% to 90% of Genia corpus. One-tailed t-Test results confirm that the ABTA system adopting our method achieves more reliable performance than the original ABTA system when the training corpus is small. The main result of this work is thus that affix features can be parametrized from small corpora at no cost in performance.

There are some aspects in which the proposed method can be improved in our future work. We are interested in investigating whether there exists a certain threshold value for the joint probability which might improve the classification accuracy of ABTA system to some extent. However, this could import supervised elements into our method. Moreover, we would like to incorporate our method into other published learning-based biological term annotation systems to see if better system performance will be achieved. However, superior parametrization will improve the annotation performance only if the affix information is not redundant with other features such as POS.

References

Ethem Alpaydin. 2004. *Introduction to Machine Learning*. MIT Press.

William B. Cavnar and John M. Trenkle. 1994. N-gram-based text categorization. In *Proc. SDAIR-94, 3rd Ann. Symposium on Doc. Analysis and Inf. Retr.*, pages 161–175, Las Vegas, USA.

Aaron Michael Cohen and William R. Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 5(1):57–71.

Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Gail Sinclair, and Christopher Manning. 2004. Exploiting context for biomedical entity recognition: From syntax to the web. In *Joint wsh. on NLP in Biomedicine and its Applications (JNLPBA-2004)*.

Kristofer Franzn, Gunnar Eriksson, Fredrik Olsson, Lars Asker Per Lidn, and Joakim Cster. 2002. Protein names and how to find them. *International Journal of Medical Informatics special issue on NLP in Biomedical Applications*, pages 49–61.

Kristofer Franzn, Gunnar Eriksson, Fredrik Olsson, Lars Asker Per Lidn, and Joakim Cster. 2003. Mining the Biomedical Literature in the Genomic Era: An Overview. *J. Comp. Biol.*, 10(6):821–855.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. 1998. Toward information extraction: Identifying protein names from biological papers. In *the Pacific Symposium on Biocomputing*, pages 707–718.

Sittichai Jiampojamarn, Nick Cercone, and Vlado Kešelj. 2005a. Automatic Biological Term Annotation Using N-gram and Classification Models. Master's thesis, Faculty of Comp.Sci., Dalhousie University.

Sittichai Jiampojamarn, Nick Cercone, and Vlado Kešelj. 2005b. Biological Named Entity Recognition using N-grams and Classification Methods. In *Conf. of the Pacific Assoc. for Computational Linguistics, PACLING'05*, Tokyo, Japan.

Ki-Joong Lee, Young-Sook Hwang, and Hae-Chang Rim. 2003. Two-phase biomedical NE recognition based on SVMs. In *Proc. of the ACL 2003 workshop on Natural language processing in biomedicine*, pages 33–40, Morristown, NJ, USA. ACL.

Donald R. Morrison. 1968. Patricia - Practical Algorithm To Retrieve Information Coded in Alphanumeric. *Journal of the ACM*, 15(4):514–534.

Goran Nenadic and Sophia Ananiadou. 2006. Mining semantically related terms from biomedical literature. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(1):22 – 43.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and novel feature sets. In *Joint wsh. on NLP in Biomedicine and its Applications (JNLPBA-2004)*.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain. In *Proc. of the ACL 2003 wsh. on NLP in Biomedicine*, pages 49–56, Morristown, NJ, USA.

Catherine Soanes and Angus Stevenson. 2004. *Oxford Dictionary of English*. Oxford University Press.

Combining Multiple Evidence for Gene Symbol Disambiguation

Hua Xu
Dept. of Biomedical Informatics, Columbia University
622 W 168th St. NY, USA
hux7002@dbmi.columbia.edu

Jung-Wei Fan
Dept. of Biomedical Informatics, Columbia University
622 W 168th St. NY, USA
fan@dbmi.columbia.edu

Carol Friedman
Dept. of Biomedical Informatics, Columbia University
622 W 168th St. NY, USA
friedman@dbmi.columbia.edu

Abstract

Gene names and symbols are important biomedical entities, but are highly ambiguous. This ambiguity affects the performance of both information extraction and information retrieval systems in the biomedical domain. Existing knowledge sources contain different types of information about genes and could be used to disambiguate gene symbols. In this paper, we applied an information retrieval (IR) based method for human gene symbol disambiguation and studied different methods to combine various types of information from available knowledge sources. Results showed that a combination of evidence usually improved performance. The combination method using coefficients obtained from a logistic regression model reached the highest precision of 92.2% on a testing set of ambiguous human gene symbols.

1 Introduction

In the past decade, biomedical discoveries and publications have increased exponentially due to high-throughput technologies such as automated genomic sequencing, and therefore, it is impossible for researchers to keep up-to-date with the most recent knowledge by manually reading the literature. Therefore, automated text mining tools, such as information retrieval and information extraction systems, have received great amounts of interest (Erhardt et al., 2006; Krallinger and Valencia, 2005). Biomedical entity recognition is a first crucial step for text mining tools in this domain, but is a very challenging task, partially due to the ambiguity (one name referring to different entities) of names in the biomedical field.

Genes are among the most important biological entities for understanding biological functions and processes, but gene names and symbols are highly ambiguous. Chen et al. (2005) obtained gene information from 21 organisms and found that ambiguities within species, across species, with English words and with medical terms were 5.02%, 13.43%, 1.10%, 2.99%, respectively, when both official gene symbols and aliases were considered. When mining MEDLINE abstracts, they found that 85.1% of mouse genes in the articles were ambiguous with other gene names. Recently, Fundel and Zimmer (2006) studied gene/protein nomenclature in 5 public databases. Their results showed that the ambiguity problem was not trivial. The degree of ambiguity also varied among different organisms. Unlike other abbreviations in the literature, which usually are accompanied by their corresponding long forms, many gene symbols occur alone without any mention of their long forms. According to Schuemie et al. (2004), only 30% of gene symbols in abstracts and 18% in full text were accompanied by their corresponding full names, which makes the task of gene symbol normalization much harder.

Gene symbol disambiguation (GSD) is a particular case of word sense disambiguation (WSD), which has been extensively studied in the domain of general English. One type of method for WSD uses established knowledge bases, such as a machine readable dictionary (Lesk, 1986; Harley and Glennon, 1997). Another type of WSD method uses supervised machine learning (ML) technolo-

BioNLP 2007: Biological, translational, and clinical language processing, pages 41–48,
Prague, June 2007.

gies (Bruce and Wiebe, 1994; Lee and Ng, 2002; Liu et al., 2002).

In the biomedical domain, there are many gene related knowledge sources, such as Entrez Gene (Maglott et al., 2005), developed at NCBI (National Center for Biotechnology Information), which have been used for gene symbol disambiguation. Podowski et al. (2004) used MEDLINE references in the LocusLink and SwissProt databases to build Bayesian classifiers for GSD. A validation on MEDLINE documents for a set of 66 human genes showed most accuracies were greater than 90% if there was enough training data (more than 20 abstracts for each gene sense).

More recently, information retrieval (IR) based approaches have been applied to resolve gene ambiguity using existing knowledge sources. Typically, a profile vector for each gene sense is built from available knowledge source(s) and a context vector is derived from the context where the ambiguous gene occurs. Then similarities between the context vector and candidate gene profile vectors are calculated, and the gene corresponding to the gene profile vector that has the highest similarity score to the context vector is selected as the correct sense. Schijvenaars et al. (2005) reported on an IR-based method for human GSD. It utilized information from either Online Mendelian Inheritance in Man (OMIM) annotation or MEDLINE abstracts. The system achieved an accuracy rate of 92.7% on an automatically generated testing set when five abstracts were used for the gene profile. Xu et al. (2007) studied the performance of an IR-based approach for GSD for mouse, fly and yeast organisms when different types of information from different knowledge sources were used. They also used a simple method to combine different types of information and reported that a highest precision of 93.9% was reached for a testing set of mouse genes using multiple types of information.

In the field of IR, it has been shown that combining heterogeneous evidence improves retrieval effectiveness. Studies on combining multiple representations of document content (Katzer et al., 1982), combining results from different queries (Xu and Croft, 1996), different ranking algorithms (Lee, 1995), and different search systems (Lee, 1997) have shown improved performance of retrieval systems. Different methods have also been developed to combine different evidence for IR tasks. The inference-network-based framework, developed by Turtle and Croft (1991), was able to combine different document representations and retrieval algorithms into an overall estimate of the probability of relevance. Fox et al. (1988) extended the vector space model to use sub-vectors to describe different representations derived from documents. An overall similarity between a document and a query is defined as a weighted linear combination of similarities of sub-vectors. A linear regression analysis was used to determine the value of the coefficients.

Though previous related efforts (Schijvenaars et al., 2005, Xu et al., 2007) have explored the use of multiple types of information from different knowledge sources, none have focused on development of formal methods for combining multiple evidence for the GSD problem to optimize performance of an IR-based method. In this study, we adapted various IR-based combination models specifically for the GSD problem. Our motivation for this work is that there are diverse knowledge sources containing different types of information about genes, and the amount of such information is continuously increasing. A primary source containing gene information is MEDLINE articles, which could be linked to specific genes through annotation databases. For example, Entrez Gene contains an annotated file called "gene2pubmed", which lists the PMIDs (PubMed ID) of articles associated with a particular gene. From related MEDLINE articles, words and different ontological concepts can be obtained and then be used as information associated with a gene. However they could be noisy, because one article could mention multiple genes. Another type of source contains summarized annotation of genes, which are more specific to certain aspects of genes. For example, Entrez Gene contains a file called "gene2go". This file lists genes and their associated Gene Ontology (GO) (Ashburner et al., 2000) codes, which include concepts related to biological processes, molecular functions, and cellular components of genes. Therefore, methods that are able to efficiently combine the different types of information from the different sources are important to explore for the purpose of improving performance of GSD systems. In this paper, we describe various models for combining different types of information from MEDLINE abstracts for IR-based GSD systems. We also evaluated the combination models using two data sets containing ambiguous human genes.

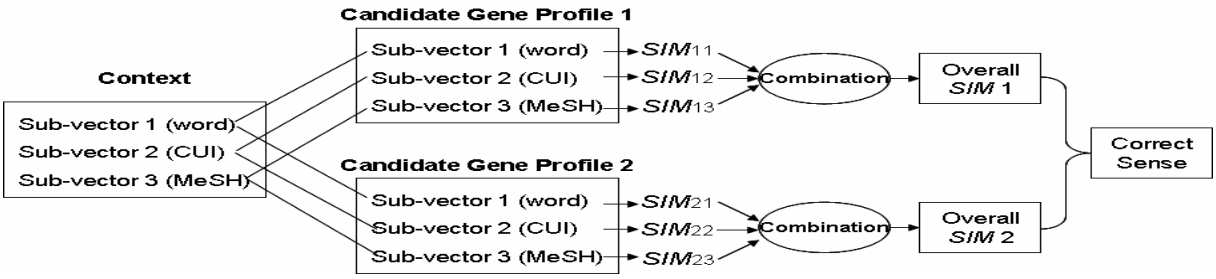


Figure 1 Overview of an IR combination-based gene symbol disambiguation approach using different types of information.

2 Methods

In this paper, we extend the IR vector space model to be capable of combining different types of gene related information in a flexible manner, thus improving the performance of an IR-based GSD system. Figure 1 shows an overview of the IR combination-based approach. We generated three different sub-vectors for the context and three for the profile, so that each sub-vector corresponded to a different type of information. The similarity scores between context and profile were measured for each type of sub-vector and then combined to generate the overall similarity scores to determine the correct sense. We explored five different combination methods using two testing sets.

2.1 Knowledge Sources and Available Information

The "gene2pubmed" file in Entrez Gene was downloaded in January 2006. A profile was then built for each gene using information derived from the related articles. We used the following three types of information: 1) Words in the related MEDLINE articles (title and abstract). This is the simplest type of information about a gene. General English stop words were removed and all other words were stemmed using the Porter stemming algorithm (Porter, 1980). 2) UMLS (Unified Medical Language System) (Bodenreider 2004) CUIs (Concept Unique Identifier), which were obtained from titles and abstracts of MEDLINE articles using an NLP system called MetaMap (Aronson 2001). 3) MeSH (Medical Subject Headings) terms, which are manually annotated by curators based on full-text articles at the National Library of Medicine (NLM) of the United States.

2.2 Document Set and Testing Sets

Using the "gene2pubmed" file, we downloaded the MEDLINE abstracts that were known to be related to human genes. Articles associated with more than 25 genes (as determined by our observation) were excluded, since they mostly discussed high-throughput technologies and provided less valuable information for GSD. This excluded 168 articles and yielded a collection of 116,929 abstracts, which were used to generate gene profiles and one of the test sets. Two test sets were obtained for evaluating the combination methods: testing set 1 was based on the "gene2pubmed" file, and testing set 2 was based on the BioCreAtIvE II evaluation.

Testing set 1 was automatically generated from the 116,929 abstracts, using the following 3 steps:

1) Identifying ambiguous gene symbols in the abstracts. This involved processing the entire collection of abstracts using an NLP system called BioMedLEE (Biomedical Language Extracting and Encoding System) (Lussier et al. 2006), which was shown to identify gene names/symbols with high precision when used in conjunction with GO annotations. When an ambiguous gene was identified in an article, the candidate gene identifiers (GeneID from Entrez Gene) were listed by the NLP system, but not disambiguated. For each ambiguous gene that was detected, a pair was created consisting of the PMID of the article and the gene symbol, so that each pair would be considered a possible testing sample. Repeated gene symbols in the same article were ignored, because we assumed only one sense per gene symbol in the same article. Using this method, 69,111 PMID and ambiguous human gene symbol pairs were identified from the above collection of abstracts.

2) Tagging the correct sense of the ambiguous gene symbols. The list of candidate PMID/gene symbol pairs generated from the articles was then compared with the list of gene identifiers known to be associated with the articles based on "gene2pubmed". If one of the candidate gene senses matched, that gene sense was assumed to be the correct sense. Then the PMID/gene-symbol pair was tagged with that sense and set aside as a testing sample. We identified a pool of 12,289 testing samples, along with the corresponding tagged senses.

3) Selecting testing set 1. We randomly selected 2,000 testing samples from the above pool to form testing set 1.

Testing set 2 was derived using the training and evaluation sets of the BioCreAtIvE II Gene Normalization (GN) task (Morgan 2007). The BioCreAtIvE II GN task involved mapping human gene mentions in MEDLINE abstracts to gene identifiers (Entrez Gene ID), which is a broader task than the GSD task. However, these abstracts were useful for creating a testing set for GSD, because whenever a gene mention mapped to more than one identifier, disambiguation was required. Therefore, it was possible to derive a list of ambiguous gene symbols based on data that was provided by BioCreAtIvE. We combined both manually annotated training (281 abstracts) and evaluation (262 abstracts) sets provided by BioCreAtIvE. Using the same process as described in step 1 of testing set 1, we processed the abstracts and identified 217 occurrences of ambiguous gene symbols from the combined set. Following a similar procedure as was used for step 2 in the testing set 1 (except that the reference standard in this case was the manually annotated results obtained from BioCreAtIvE instead of "gene2pubmed"), we obtained 124 PMID/gene-symbol pairs with the corresponding tagged senses, which formed testing set 2.

Because one article may contain multiple ambiguous gene symbols, a total of 2,048 PMIDs were obtained from both testing sets 1 and 2. Articles with those PMIDs were excluded from the collection of 116,929 abstracts. We used the remaining document set to generate gene profiles, which were used for both testing sets.

2.3 Profile and Context Vectors

For each gene in "gene2pubmed" file, we created a profile. It consisted of three sub-vectors containing *word*, *CUI,* or *MeSH,* respectively, using the information derived from the related MEDLINE abstracts. Similarly, a context vector was also formed for each testing sample, using three sub-vectors containing *word*, *CUI,* or *MeSH*, which were derived from the abstract whose PMID was stated in the testing sample. The tf-idf weighting schema (Salton and Buckley, 1988) was used to assign weights to index terms in the profile and context sub-vectors. Given a document *d*, the Term Frequency (tf) of term *t* is defined as the frequency of *t* occurring in *d*. The Inverse Document Frequency (idf) of term *t* is defined as the logarithm of the number of all documents in the collection divided by the number of documents containing the term *t*. Then term *t* in document *d* is weighted as tf*idf.

2.4 Similarity Measurement

The similarity score between the same type of context and profile sub-vectors were measured as cosine similarity of two vectors. The cosine similarity between two vectors *a* and *b* is defined as the inner product of *a* and *b*, normalized by the length of two vectors. See the formula below:

$$\text{Sim}(a,b) = \text{cosine } \theta = \frac{a \cdot b}{|a||b|} \quad \text{where}$$

$$|a| = \sqrt{a_1^2 + a_2^2 + \ldots + a_n^2} \qquad |b| = \sqrt{b_1^2 + b_2^2 + \ldots + b_n^2}$$

We built three basic classifiers that used only one type of sub-vector: *word*, *CUI,* or *MeSH,* respectively, recorded three individual similarity scores of each sub-vector for each candidate gene of all testing samples. We implemented five methods to combine similarity scores from each basic classifier, which are described as follows:

1) *CombMax* - Each individual similarity score from a basic classifier was normalized by dividing the sum of similarity scores of all candidate genes for that basic classifier. Then the decision made by the classifier with the highest normalized score was selected as the final decision of the combined method.
2) *CombSum* - Each individual similarity score from a basic classifier was normalized by dividing the maximum similarity score of all candidate genes for that basic classifier. The overall similarity score of a candidate gene was considered to be the sum of the normalized similarity scores from all three basic classifiers for that gene. The candidate gene

with the highest overall similarity was selected as the correct sense.

3) *CombSumVote* - The overall similarity score was considered as the similarity score from *CombSum,* multiplied by the number of basic classifiers that voted for that gene as the correct sense.

4) *CombLR* - The overall similarity score was defined as a predicted probability (*P*) of being the correct sense, given the coefficients obtained from a logistic regression model and similarity scores from all three basic classifiers for that gene. The relation between dependent variable (probability of being the correct sense) and independent variables (similarity scores from individual basic classifiers) of the logistic regression model is shown below, where *Cs* (C_{word}, C_{cui}, C_{mesh} and *C)* are the coefficients, and *SIMs* (SIM_{word}, SIM_{cui}, SIM_{mesh}) are the individual similarity scores from the basic classifiers. To obtain the model, we divided 2,000 testing samples into a training set and a testing set, as described in section 2.5. For samples in the training set, the correct gene senses were labeled as "1" and incorrect gene senses were labeled as "0". Then logistic regression was applied, taking the binary labels as the value of the dependent variable and the similarities from the basic classifiers as the independent variables. In testing, coefficients obtained from training were used to predict each candidate gene's probability of being the correct sense for a given ambiguous symbol.

$$P = \frac{e^{Cword*SIMword+Ccui*SIMcui+Cmesh*SIMmesh+C}}{1+e^{Cword*SIMword+Ccui*SIMcui+Cmesh*SIMmesh+C}}$$

5) *CombRank* – Instead of using the similarity scores, we ranked the similarity scores and used the rank to determine the combined output. Following a procedure called Borda count (Black, 1958), the top predicted gene sense was given a ranking score of N-1, the second top was given N-2, and so on, where N is the total number of candidate senses. After each sense was ranked for each basic classifier, the combined ranking score of a candidate gene was determined by the sum of ranking scores from all three basic classifiers. The sense with the highest combined ranking score was selected as the correct sense.

2.5 Experiments and Evaluation

In this study, we measured both precision and coverage of IR-based GSD approaches. Precision was defined as the ratio between the number of correctly disambiguated samples and the number of total testing samples for which the disambiguation method yielded a decision. When a candidate gene had an empty profile or different candidate gene profiles had the same similarity scores (e.g. zero score) with a particular context vector, the disambiguation method was not able to make a decision. Therefore, we also reported on coverage, which was defined as the number of testing samples that could be disambiguated using the profile-based method over the total number of testing samples. We evaluated precision and coverage of different combined methods for gene symbol disambiguation on both testing sets.

Results of three basic classifiers that used a single type of information were reported as well. We also defined a baseline method. It used the majority sense of an ambiguous gene symbol as the correct sense. The majority sense is defined as the gene sense which was associated with the most MEDLINE articles based on the "gene2pubmed" file.

To evaluate the *CombLR*, we used 10-fold cross validation. We divided the sense-tagged testing set into 10 equal partitions, which resulted in 200 testing samples for each partition. When one partition was used for testing, the remaining nine partitions were combined and used for training, which also involved deriving coefficients for each round. To make other combination methods comparable with *CombLR*, we tested the performance of other combination methods on the same partitions as well. Therefore, we had 10 measurements for each combination method. Mean precision and mean coverage were reported for those 10 measurements. For testing set 2, we did not test the *CombLR* method because the set was too small to train a regression model.

We used Friedman's Test (Friedman, 1937) followed by Dunn's Test (Dunn, 1964), which are non-parametric tests, to assess whether there were significant differences in terms of median precision among the different single or combined methods.

3 Results

Results of different combination methods for testing set 1 are shown in Table 1, which contains the mean precision and coverage for 10-fold cross validation, as well as the standard errors in parentheses. All IR-based gene symbol disambiguation approaches showed large improvements when compared to the baseline method. All of the combination methods showed improved performance when compared to results from any run that used a single type of information. Among the five different combination methods, *CombLR* achieved the highest mean precision of 0.922 for testing set 1. *CombSum,* which is a simple combination method, also had a good mean precision of 0.920 on testing set 1. The third Column of Table 1 shows that coverage was in a range of 0.936-0.938.

Run	Precision	Coverage
Baseline	0.707 (0.032)	0.992 (0.005)
Word	0.882 (0.023)	0.937 (0.017)
CUI	0.887 (0.022)	0.938 (0.017)
MeSH	0.900 (0.021)	0.936 (0.017)
CombMax	0.909 (0.020)	0.938 (0.017)
CombSum	0.920 (0.019)	0.937 (0.017)
CombSumVote	0.917(0.019)	0.938 (0.017)
CombLR	**0.922** (0.019)	0.938 (0.017)
CombRank	0.918 (0.020)	0.938 (0.017)

Table 1. Results on testing set 1.

Run	Precision	Coverage
Baseline	0.593	0.991
Word	0.872	0.944
CUI	0.897	0.944
MeSH	0.863	0.944
CombMax	**0.906**	0.944
CombSum	**0.906**	0.944
CombSumVote	0.897	0.944
CombRank	0.889	0.944

Table 2. Results on testing set 2.

We performed Friedman's test followed by Dunn's test on each single run: *word*, *CUI* or *MeSH,* with all combination runs respectively. Friedman tests showed that differences of median precisions among the different methods were statistically significant at α=0.05. Dunn tests showed that combination runs *CombSum*, *CombSumVote, CombLR,* and *CombRank* were statistically significantly better than single runs using *word* or *CUI*. For single run using *MeSH*, combination runs *CombLR* and *CombSum* were statistically significantly better.

The results of different runs on testing set 2 are shown in Table 2. Most combined methods, except *CombRank*, showed improved precision. The highest precision of 0.906 was reached when using *CombSum* and *CombMax* methods. Note that the logistic regression method was not applicable. The coverage for testing set 2 was 0.944 for all of the methods.

4 Discussion

4.1 Why Combine?

As stated in Croft (2002), a Bayesian probabilistic framework could provide the theoretical justification for evidence combination. Additional evidence with smaller errors can reduce the effect of large errors from one piece of evidence and lower the average error.

The idea behind *CombMax* was to use the single classifier that had the most confidence, but it did not seem to improve performance very much because it ignored evidence from the other two basic classifiers. The *CombSum* was a simple combination method, but with reasonable performance, which was also observed by other studies for the IR task (Fox and Shaw, 1994). *CombSumVote* was a variant of *CombSum*. It favors the candidate genes selected by more basic classifiers. In Lee (1997), a similar implementation of *CombSumVote* (named "CombMNZ") also achieved better performance in the IR task. *CombLR*, the combination method trained on a logistic regression model, achieved the best performance in this study. It used a set of coefficients derived from the training data when combining the similarities from individual basic classifiers. Therefore, it could be considered as a more complicated linear combination model than *CombSum*. In situations where training data is not available, *CombSum* or *CombSumVote* would be a good choice. *CombRank* did not perform as well as methods that used similarity scores, probably due to the loss of subtle probability information in the similarity scores. We explored ranking because it was independent of the weighting schema and could be valuable if it performed well.

The typical scenario where combination should help is when a classifier based on one type of information made a wrong prediction, but the other(s), based on different types of information, made the correct predictions. In those cases, the overall prediction may be correct when an appropriate combination method applies. For example, an ambiguous gene symbol *PDK1* (in the article with PMID 10856237), which has two possible gene senses ('GeneID:5163 pyruvate dehydrogenase kinase, isoenzyme 1' and 'GeneID:5170 3-phosphoinositide dependent protein kinase-1'), was incorrectly predicted as 'GeneID: 5163' when only "word" was used. But the classifiers using "CUI" and "MeSH" predicted it correctly. When the *CombSum* method was used to combine the similarity scores from all three classifiers, the correct sense 'GeneID: 5170' was selected. When all three classifiers were incorrect in predicting a testing sample, generally none of the combination methods would help in making the final decision correct. Therefore, there is an upper bound on the performance of the combined system. In our case, we detected that all three classifiers made incorrect predictions for 65 testing samples of the 2,000 samples. Therefore, the upper bound would be 1,935/2,000=96.7%.

The methods for combining different types of information from biomedical knowledge sources described in this study, though targeted to the GSD problem, could be also applicable to other text mining tasks that are based on similarity measurement, such as text categorization, clustering, and the IR task in the biomedical domain.

4.2 Coverage of the Methods

The IR-based gene symbol disambiguation method described in this paper aims to resolve intra-species gene ambiguity. We focused on ambiguous gene symbols within the human species and used articles known to be associated with human genes. Fundel and Zimmer (2006) reported that the degree of ambiguity of the human gene symbols from Entrez Gene was 3.16%–3.32%, which is substantial. However, this is only part of the gene ambiguity problem.

Based on the "gene_info" file downloaded in January 2006 from Entrez Gene, there were a total of 32,852 human genes. Based on the "gene2pubmed" file, 24,170 (73.4%) out of 32,852 human genes have at least one associated MEDLINE article, which indicates that profiles could be generated for at least 73.4% of human genes. On average, there are 9.02 MEDLINE articles associated with a particular human gene. Coverage reported in this study was relatively high because the testing samples were selected from annotated articles as listed in "gene2pubmed", and not randomly from the collection of all MEDLINE abstracts.

4.3 Evaluation Issues

It would be interesting to compare our work with other related work, but that would require use of the same testing set. For example, it is not straightforward to compare our precision result (92.2%) with that (92.7%) reported by Schijvenaars et al. (2005), because they used a testing set that was generated by removing ambiguous genes with less than 6 associated articles for each of their senses, and they did not report on coverage. The data set from the BioCreAtIvE II GN task therefore is a valuable testing set that enables evaluation and comparison of other gene symbol disambiguation methods. From the BioCreAtIvE abstracts, we identified 217 occurrences of ambiguous gene symbols, but only 124 were annotated in the BioCreAtIvE data set. There are a few possible explanations for this. First, the version of the Entrez Gene database used by the NLP system was not the most recent one, so some new genes were not listed as possible candidate senses. The second issue is related to gene families or genes/proteins with multiple sub-units. According to the 'gene_info' file, the gene symbol "IL-1" is a synonym for both "GeneID: 3552 interleukin 1, alpha" and "GeneID: 3553 interleukin 1, beta". Therefore, the NLP system identified it as an ambiguous gene symbol. When annotators in the BioCreAtIvE II task saw a gene family name that was not clearly mapped to a specific gene identifier in Entrez Gene, they may not have added it to the mapped list. In Morgan et al. (2007), it was suggested that mapping gene family mentions might be appropriate for those entities. Testing set 2 was a small set and results from that set might not be statistically meaningful, but it is useful for comparing with others working on the same data set.

In this paper, we focused on the study of improvements in precision of the gene symbol disambiguation system. When combining information from different knowledge sources, coverage may

also be increased by benefiting from the cross-coverage of different knowledge sources.

5 Conclusion and Future Work

We applied an IR-based approach for human gene symbol disambiguation, focusing on a study of different methods for combining various types of information from available knowledge sources. Results showed that combination of multiple evidence usually improved the performance of gene symbol disambiguation. The combination method using coefficients obtained from a logistic regression model reached the highest precision of 92.2% on an automatically generated testing set of ambiguous human gene symbols. On a testing set derived from BioCreAtIvE II GN task, the combination method that performed summation of individual similarities reached the highest precision of 90.6%. However, the regression-based method could not be used, because the testing sample was small.

In the future, we will add information that is specifically related to genes, such as GO codes, into the combination model. Meanwhile, we will also study the performance gain in terms of coverage by integrating different knowledge sources.

Acknowledgements

This work was supported in part by Grants R01 LM7659 and R01 LM8635 from the National Library of Medicine, and Grant NSF-IIS-0430743 from the National Science Foundation. We would like to thank Alexander Morgan for providing the evaluation set from the BioCreAtIvE II GN task.

References

Aronson, A. R. 2001. *Proc. AMIA. Symp.*, 17-21.

Ashburner, M. et al. 2000. *Nat Genet*, 25, 25-29.

Black, D. 1958. *Cambridge University Press.*

Bodenreider, O. 2004. *Nucleic Acids Research*, 2004, 32, D267-D270.

Bruce, R. and Wiebe, J. 1994. *Proceedings of ACL 1994*, 139-146.

Chen, L., Liu, H. and Friedman, C. 2005. *Bioinformatics*, 21, 248-256.

Croft, W. 2002. *Advances in Information Retrieval.* Springer Netherlands, Chapter 1, 1-36

Dunn, O. J. 1964. *Technometrics*, 6, 241-252.

Erhardt, R.A., Schneider, R. and Blaschke, C. 2006. *Drug Discov. Today*, 11, 315-325.

Fox, E., Nunn, G., and Lee, W. 1988. *Proceedings of the 11th ACM SIGIR Conference* on Research and Development in Information Retrieval, 291–308.

Fox, E. and Shaw, J. 1994. *Proceedings TREC-2*, 243–252.

Friedman, M. 1937. *Journal of the American Statistical Association*, 32, 675-701.

Fundel, K. and Zimmer, R. 2006. *BMC. Bioinformatics.*, 7: 372.

Harley, A. and Glennon, D. 1997. *Proc. SIGLEX Workshop "Tagging Text With Lexical Semantics"*, 74-78.

Katzer, J., McGill, M., Tessier, J., Frakes,W., and DasGupta, P. 1998. *Information Technology: Research and Development*, 1(4):261–274.

Krallinger, M. and Valencia, A. 2005. *Genome Biol.*, 6, 224.

Lee, J. 1995. *Proceedings of the 18th ACMSIGIR Conference on Research and Development in Information Retrieval*, 180–188.

Lee, J. 1997. *Proceedings of the 20th ACM SIGIR Conference on Research and Development in Information Retrieval*, 267–276.

Lee, Y. K. and Ng, H. T. 2002. *Proc EMNLP 2002,* 41-48.

Lesk, M. 1986. *1986 SIGDOC Conference*, 24-26.

Liu, H., Johnson, S. B. and Friedman, C. 2002. *J. Am. Med. Inform. Assoc.*, 9, 621-636.

Lussier, Y., Borlawsky, T., Rappaport, D., Liu, Y., Friedman, C. 2006. *Pac. Symp. Biocomput.*, 11, 64-75.

Maglott D, Ostell J, Pruitt KD, Tatusova T. 2005. *Nucleic Acids Res.*, 3, D54-D58.

Morgan, A., Wellner, B., Colombe, J. B., Arens, R., Colosimo, M. E., Hirschman L. 2007. *Pacific Symposium on Biocomputing* 12:281-291.

Podowski, R.M., Cleary, J.G., Goncharoff, N.T., Amoutzias, G., Hayes W.S. 2004. *Proc IEEE Comput Syst Bioinform Conf*, *2004*, 415-24.

Porter,M.F. 1980. *Program*, 14, 130-137.

Salton, G. and Buckley, C. 1988. *Information Processing & Management*, 24, 513-523.

Schijvenaars, B.JA. et al. 2005. *BMC. Bioinformatics.*, 6:149.

Schuemie, M.J. et al. 2004. *Bioinformatics*, 20, 2597-2604.

Turtle, H. and Croft, W. 1991. *ACM Transactions on Information Systems*, 9(3):187–222.

Xu, H., Fan, J. W., Hripcsak, G., Mendonça A. E., Markatou, M., Friedman, C. 2007. *Bioinformatics*, doi: 10.1093/bioinformatics/btm056

Xu, J. and Croft,W. 1996. Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, pages 4–11.

Mining a Lexicon of Technical Terms and Lay Equivalents

Noemie Elhadad and **Komal Sutaria**
Computer Science Department
The City College of New York
New York, NY 10031
noemie@cs.ccny.cuny.edu, kdsutaria@gmail.com

Abstract

We present a corpus-driven method for building a lexicon of semantically equivalent pairs of technical and lay medical terms. Using a parallel corpus of abstracts of clinical studies and corresponding news stories written for a lay audience, we identify terms which are good semantic equivalents of technical terms for a lay audience. Our method relies on measures of association. Results show that, despite the small size of our corpus, a promising number of pairs are identified.

1 Introduction

The field of health literacy has garnered much attention recently. Studies show that most documents targeted at health consumers are ill-fitted to the intended audience and its level of health literacy (Rudd et al., 1999; McCray, 2005). While there are many components involved in health literacy that are specific to the reader (e.g., reading level and cultural background), we investigate what can be done from the standpoint of the text to adapt it to the literacy level of a given reader. As such, we set ourselves in the context of a text-to-text generation system, where a technical text is edited to be more comprehensible to a lay reader. An essential resource for such an editing tool is a lexicon of paraphrases, or semantically equivalent terms. In this paper, we investigate a corpus-driven method for building such a lexicon. We focus on terms that are recognized by the UMLS (UMLS, 1995), both for technical and lay candidate terms for equivalence.

Because we have lay audiences in mind, our definition of semantic equivalence must be broader than a notion of strict medical equivalence utilized by medical experts. Thus, while a medical dictionary like UMLS assigns different concept unique identifiers (CUIs) to two particular terms, such as *percutaneous transluminal coronary angioplasty* and *angioplasty*, these terms should be considered semantically equivalent for the purposes of lay readers.

Besides enabling a text tailoring system to adapt technical texts for a lay audience, a lexicon of semantically equivalent technical/lay terms would benefit other tools as well. For instance, the Consumer Health Vocabulary initiative[1] is a comprehensive list of UMLS terms familiar to lay readers. Our lexicon could help augment the terms with equivalence links to technical terms. While much research of late has been devoted to identifying terms incomprehensible to lay readers, such research has not established links between technical terms and equivalent lay terms beyond their CUI information (Zeng et al., 2005; Elhadad, 2006).

The key points of our approach are: (1) the use of combined measures of association to identify pairs of semantically equivalent terms, and (2) a knowledge-based heuristic which acts as a powerful filter for identifying semantically equivalent pairs. Our method does not rely on human labeling of semantically equivalent term pairs. As such, it is unsupervised, and achieves results that are promising considering the small size of the corpus from which the results are derived.

This paper is organized as follows. The next section describes our parallel corpus of paired technical/lay documents. The Methods section describes the different measures of association we experimented with, how we combine them to leverage their complimentary strengths, and our semantic filter. The Results section reports the evaluation against our gold standard and a discussion of our results.

[1] http://www.consumerhealthvocab.org

BioNLP 2007: Biological, translational, and clinical language processing, pages 49–56,
Prague, June 2007.

2 Data Description

Because our ultimate goal is to learn, in a data-driven fashion, semantic equivalents of terms that are too technical for lay readers, we can benefit from having instances of texts which relay similar information but are conveyed in different styles. We collect a corpus similar in structure to those used in the field of statistical machine translation. But, instead of having two collections in different languages, we collect texts written for two different audiences: medically trained readers (technical collection) and health consumers (lay collection).

The lay collection is composed of news stories from the ReutersHealth E-line newsfeed[2] summarizing research in the medical field. Reuters journalists take technical publications and report the main findings and methods and, on occasion, include interviews with the authors of the scientific publication. The stories are targeted at a lay audience with a 12th-grade reading level. Furthermore, every story in our collection contains a reference to the original scientific publication. Thus, it is possible to gather the original texts, which convey similar information but were written for a technical audience. The stories draw upon studies from reputable medical journals, such as Annals of Internal Medicine, New England Journal of Medicine and Lancet.

The technical collection in our corpus is composed of the original scientific articles corresponding to each news story in the lay collection. Accordingly, the lay and technical collections contain the same number of documents and are parallel at the document level. That is, each technical document has a lay equivalent and vice-versa. Because a lay document is a summary of a technical article and is, hence, much shorter than the original scientific article, we decided to include only the abstract of the technical document in our collection. This way, the technical and lay documents are comparable in content and length. It should be noted, however, that the content in a technical/lay document pair is not parallel, but comparable (McEnery and Xiao, 2007): there is no natural sentence-to-sentence correspondence between the two texts. This is to be expected: technical abstracts contain many technical details, while lay stories, to provide background, introduce information entirely absent from abstracts. In addition, the lay stories drastically rearrange the order in which information is typically conveyed in technical abstracts. For these reasons, our corpus is not parallel at the sentence level and, thus, differs from other bilingual parallel corpora used in machine translation.

	Words			Sentences		
	Min	Max	Avg	Min	Max	Avg
Technical	137	565	317	5	18	10
Lay	187	1262	444	6	42	15

Table 1: Statistics for the Technical and Lay collections. Each contains 367 documents.

To ensure that some significant number of terms appears with sufficient frequency in our corpus in order to induce equivalent pairs automatically, we focused on articles and stories in a single domain: cardiology. We identified the original scientific article manually, as the lay document only contains a reference, not an actual link. For this reason, only a relatively small amount of data could be collected: 367 pairs of documents (see Table 1 for statistics).

3 Methods

3.1 Data Processing

We focus in this paper on finding term equivalents when both terms are recognized by the UMLS. Thus, our first step in processing our collections is to identify terms as defined by the UMLS. Both collections are processed by our tool TermFinder (Teufel and Elhadad, 2002). Sentences are identified and the texts are tokenized and tagged with part-of-speech information. Noun phrases are identified with a shallow parser. Next, terms are identified by looking up the noun phrases in the meta-lexicon of UMLS for an exact match. Terms are tagged with their concept unique identifier (CUI) and a semantic type, both provided by UMLS. For our purposes, we only consider a subset of all the terms listed in UMLS, based on their semantic type. This is due to the fact that certain UMLS semantic types are unlikely to yield technical terms in need of simplification. As such, terms belonging to semantic types such as "Activity," "Family Group" or "Behavior" were left untagged. Terms with semantic types such as "Disease or Syndrome" or "Therapeutic or Preven-

[2] http://www.reutershealth.com

	Corresponding lay doc. contains *lay_term*	Corresponding lay doc. does not contain *lay_term*
Technical doc. contains *tech_term*	a	b
Technical doc. does not contain *tech_term*	c	d

Table 2: Contingency table for (*tech_term*, *lay_term*).

tive Procedure," on the other hand, were considered terms. For instance, both the terms *PTCA* and *percutaneous transluminal coronary angioplasty* have the same CUI C0002997, as they are considered synonyms by UMLS. The term *balloon angioplasty* has the CUI C0002996. Both C0002997 and C0002996 have the semantic type "Therapeutic or Preventive Procedure."

3.2 Contingency Table

We call (*tech_term*, *lay_term*) a term pair, where *tech_term* is a term occurring in one or more technical documents and *lay_term* is a term present in at least one of the corresponding lay documents.[3] For any such pair, we can compute a contingency table based on co-occurrence. Our definition of co-occurrence is slightly unusual: *tech_term* and *lay_term* co-occur in one document pair if *tech_term* appears at least once in the technical document and *lay_term* appears at least once in the corresponding lay document. Our unit of content is document frequency for a CUI, *i.e.*, the number of documents in which a given CUI appears. For instance, in our data, the contingency table for the term pair *(MI, heart attack)* shows the following counts: the document frequency of the CUI corresponding to *MI* in the technical collection is 98; the document frequency of the CUI corresponding to *heart attack* in the lay collection is 161. Among these documents, there are 84 technical/lay document pairs (out of the total of 367 paired documents) in which the CUI for *MI* occurs on the technical side and the CUI for *heart attack* occurs on the lay side. Hence, the contingency table for this term pair is, following the notations of Table 2: a = 84, b = 98-84 = 14, c = 161-84 = 77, and d = 367-98-161+84 = 192.

At this stage of processing, lexical terms are abstracted by their CUIs. We do this to maximize the possible evidence that two terms co-occur. For instance, the document frequency for *MI* in our technical collection is 20, while the document frequency for its corresponding CUI is 98. Section 3.7 describes how we proceed from identifying equivalent terms at the CUI level to finding lexical equivalents.

3.3 Gold Standard

To evaluate the validity of our approach, we collected all possible term pairs at the CUI level in our corpus (that is, all the term pairs for which a contingency table is computed). We then whittled this set down to those pairs where each CUI occurs in at least two documents. This resulted in 2,454 pairs of CUIs. We asked our medical expert, an internist in practice who interacts with patients on a daily basis, to indicate for each pair whether the terms were equivalent from a medical standpoint *in the context of communicating with a patient*.[4] An operational test for testing the equivalence of two terms is whether he would use one term for the other when talking to a patient. We indicated to our expert that the terms should be equivalent *out of context*. So, for instance, while the pair (myocardial infarction, complication) could be deemed equivalent in certain specific contexts, these terms are not generally considered equivalent. Table 3 shows examples of pairs annotated as semantic equivalents for lay readers.[5] The list of terms contained only the actual lexical terms and no information from the UMLS to avoid biasing our expert.

Out of the 2,454 CUI pairs provided to our medical expert, 152 pairs were labeled as equivalent. Out of the 152 pairs, only 8 (5.3%) had different semantic types. Interestingly, 84 pairs (55.3%) had different CUIs. This confirms our intuition that the notion of semantic equivalence for lay readers is looser than for medically knowledgeable readers.

[3] This means that if *tech_term* and *lay_term* have no technical/lay document in common, *lay_term* is not considered a possible candidate for semantic equivalence for *tech_term*.

[4] While it is in some ways counterintuitive to rely on a technical expert to identify lay semantic equivalents, this expertise helps us validate equivalences from a medical standpoint.

[5] In the table, DIGN stands for "Diagnostic Procedure," DISS for "Disease or Symptom," FIND for "Finding," and PATH for "Pathological Finding."

Technical term	Lay term
myocardial infarction | C0027051 | DISS	heart attack | C0027051 | DISS
SBP | C0428880 | DIGN	systolic blood pressure | C0428880 | DIGN
atrial fibrillation | C0004238 | PATH	arrhythmia | C0003811 | PATH
hypercholesterolemia | C0020443 | DISS	high cholesterol | C0848569 | FIND
mental stress | C0038443 | DISS	stress | C0038435 | PATH

Table 3: Examples from the gold standard of term pairs considered equivalent.

3.4 Measures of Association

Given a term pair (*tech_term*, *lay_term*) and its corresponding contingency table, we want to determine whether *lay_term* is a valid semantic equivalent of *tech_term* from the standpoint of a lay reader. We rely on three alternative measures of association introduced in the Statistics literature: the χ^2 statistic, the λ measure, and odds ratio. All of these measures are computed as a function of the contingency table, and do not rely on any human labeling for equivalence. Measures of association have been used traditionally to identify collocations (Manning and Schütze, 1999). Here we investigate their use for building a lexicon.

3.4.1 The χ^2 Statistic

The standard chi-square statistic (χ^2) is used to determine whether the deviation of observed data from an expected event occurs solely by chance (Goodman and Kruskal, 1979). Our null hypothesis for this task is that the presence of *lay_term* in a lay document is independent of the presence of *tech_term* in its correspondent technical document. Thus, any pair of terms for which the χ^2 is above the critical value at a given level of significance are considered semantic equivalents. One important constraint for the measures to be valid is that the observed data be large enough (more than five observations per cell in the contingency table).

The χ^2 statistic for our 2x2 contingency table, and with N being the total number of document pairs, is calculated as follows:

$$\chi^2 = \frac{N(ad-bc)^2}{(a+b)(a+c)(c+d)(b+d)}$$

Since χ^2 is a true statistic, we can rely on critical values to filter out pairs with low associative power. In our case, we set the significance level at .001 (with a critical value for χ^2 of 10.83).

	C0011847	¬ C0011847	Sum
C0011849	a = 13	b = 8	21
¬ C0011849	c = 40	d = 306	346
Sum	53	314	367

Table 4: Contingency table for (C0011849, C0011847).

3.4.2 The λ and λ* Measures

The lambda measure (λ) assesses the extent to which we can predict the presence of *lay_term* in a lay document by knowing whether the original technical document contained *tech_term* (Goodman and Kruskal, 1979). λ is an asymmetrical measure of association. Since a lay document is always written based on an original technical document, it is a plausible assumption that the presence of a specific term in the technical document influenced the lexical choices of the author of the lay document. Thus, we consider the presence of *tech_term* in a technical document the antecedent to the presence of *lay_term* in the corresponding lay document, and, accordingly, operate in the setting of predicting the presence of *lay_term*.

We present the intuition behind λ in the context of the following example. Consider the contingency table for the technical CUI C0011849 (*diabetes mellitus*) and C0011847 (*diabetes*) in Table 4. The task is, given a random lay document, to predict which of two available categories it belongs to: either it contains the lay CUI (in our example, CUI C0011847 for *diabetes*) or it does not. There are two possible cases: either (1) we do not have any knowledge about the original technical document, or (2) we know the original technical document and, therefore, we know whether it contains the antecedent (in our example, CUI C0011849 for *diabetes mellitus*).

Without any prior knowledge (case (1)), the safest prediction we can make about the lay document is the category with the highest probabil-

ity. The probability of error in case (1) is $P_{err1} = \frac{N-Max(a+c,b+d)}{N}$.
In our example, the safest bet is ¬ C0011847, with a raw count of 314 documents, and a probability of error of $P_{err1} = 0.1444$.

If we have prior knowledge about the original technical document (case (2)), then our safest prediction differs. If we know that the technical document contains the CUI C0011849 (*diabetes mellitus*), then our safest prediction is the category with the highest probability: C0011847, with a raw count of 13 documents. If, on the other hand, we know that the technical document does not contain the CUI C0011849, our safest prediction is the category ¬ C0011847, with a raw count of 306 documents. Thus, overall the probability of error in case (2) is $P_{err2} = \frac{N-(Max(a,b)+Max(c,d))}{N}$.
In our example, knowledge about the original technical document lowers the probability of error to $P_{err2} = 0.1308$.

The λ measure is defined as the relative decrease in probability of error in guessing the presence of *lay_term* in a lay document $\lambda = \frac{P_{err1}-P_{err2}}{P_{err1}}$ which, using our notation for contingency tables, can be expressed as

$$\lambda = \frac{Max(a,b) + Max(c,d) - Max(a+c, b+d)}{N - Max(a+c, b+d)}$$

In our example, $\lambda = 0.094$. λ ranges between 0 and 1. A value of 1 means that knowledge about the presence of *tech_term* in the original technical document completely specifies the presence of *lay_term* in its corresponding lay document. A value of 0 means that knowledge about the presence of *tech_term* in the original technical document does not help in predicting whether *lay_term* is present in its corresponding lay document.

The λ measure is not a test of significance like χ^2. For instance, while two independent variables necessarily have a λ of 0, the opposite is not necessarily true: it is possible for two dependent variables to have a λ of 0. In our setting in particular, any contingency table where a=b will provide a λ of 0.

Since λ is computed as a function of maxima of rows and columns, λ can easily be biased toward the original proportions in the antecedent. In our example, for instance, a very large proportion of technical documents has no occurrence of C0011849, *diabetes mellitus* (94.3% of the technical documents). But for our purposes, such contingencies should not affect our measure of association, as the proportion of technical documents happening not to contain a particular term is just an artificial consequence of corpus collection. λ* is a variant of λ also proposed by Goodman and Kruskal (1979) and is able to take this fact into account. It is computed using the same formula as λ, but the elements of the contingency table are modified so that each category of the antecedent is equally likely. In our case, this means: N*=1, a*=0.5a/N(a+b), b*=0.5b/N(a+b), c*=0.5c/N(c+d), and d*=0.5d/N(c+d). Going back to our example of *diabetes mellitus* and *diabetes*, we now find $\lambda* = 0.324$, which is much higher than the original λ of 0.094, and which indicates a strong association.

We focus on λ* as a measure of association for semantic equivalence of term pairs. Since λ and λ* are not true statistics, there is no significance level we can rely on to set a threshold for them. Instead, we estimate an optimal threshold from the performance of λ* on a development set. The development set was obtained in the same manner as the gold standard and contains 50 term pairs. This is a small number of pairs, but the term pairs in the development set were carefully chosen to contain mostly semantically equivalent pairs. In our experiments, the optimal value for λ* was 0.3. Thus, λ* is used as a binary test for our purposes: *tech_term* and *lay_term* are considered semantically equivalent if their λ* is above 0.3.

3.4.3 Odds Ratio

Odds ratio is a measure of association that focuses on the extent to which one category in the contingency table affects another (Fleiss et al., 2003). For our contingency table, the odds ratio is expressed as follows:

$$OR = \frac{ad}{bc}$$

For instance, given the contingency table of Table 4, the odds ratio for the pair *(diabetes mellitus, diabetes)* is 12.43, which means that a lay document is 12.43 times more likely to contain the CUI C0011847, for *diabetes*, if its original technical document contains the term C0011849, for *diabetes mellitus*.

Like λ*, odds ratio is not a true statistic and, therefore, does not have any critical value for statistical significance. We estimated the optimal value of a threshold for OR based on the same development set described above. The threshold for OR is set to 6. Thus, OR is used as a binary test for our purposes: *tech_term* and *lay_term* are considered semantically equivalent if their OR is above 6.

3.5 Combining the Measures of Association

Each of the measures of association described above leverages different characteristics of the contingency tables, and similarly, each has its limitations. For instance, χ^2 cannot be computed when there are not sufficient observations, and λ* can equal 0, even when there is a strong association between the two terms. We combine measures of association in the following fashion: two terms are considered equivalent if at least one of the measures determined so.

3.6 Semantic Filtering

The measures of association described above and their combination provide information solely based on corpus-derived data. Since all our counts are based on co-occurrence, a measure of association by itself can encompass many types of semantic relations. For instance, the pair for (*stroke, brain*) tests positive with our three measures of association. Indeed, there is a strong semantic association between the two terms: strokes occur in the brain. These terms, however, do not fit our definition of semantic equivalence.

We rely on knowledge provided by the UMLS, namely semantic types, to help us filter equivalent types of associations among the candidate term pairs. One can assume that sharing semantic types is a necessary condition for semantic equivalence. Our semantic filter consists of testing whether *tech_term* and *lay_term* share the same semantic types, as identified by our tool TermFinder.

3.7 Lexical Choice

So far, term pairs are at the CUI level. The measures of association and the semantic filter provide a way to identify candidates for semantic equivalence. We still have to figure out which particular lexical items among the different lexical terms of a given CUI are appropriate for a lay reader. For instance, the pair (C0027051, C0027051) is considered a good candidate for semantic equivalence. In the technical collection, the lexical terms contributing to the CUI are *AMI*, *AMIs*, *MI*, *myocardial infarction*, *myocardial infarct* and *myocardial necrosis*. In the lay collection, however, the lexical terms contributing to the same CUI are *heart attack*, *heart attacks*, and *myocardial infarction*. Clearly, not all lexical items for a given CUI are appropriate for a lay reader.

To select an appropriate lay lexical term, we rely on the term frequency of each lexical item in the lay collection (Elhadad, 2006). In our example, the lexical term "heart attack" has the highest term frequency in the lay collection among all the variants with the same CUI. Thus, we chose it as a semantic equivalent of any lexical term of the CUI C0027051 in the technical collection.

If a technical term has several candidate semantic equivalents at the CUI level, the lexical lay term is chosen among all the lay terms. For instance, *(adverse effect, side effect)* and *(adverse effect, complications)* are two valid equivalents, but *side effects* has a term frequency of 16 in our lay collection, and *complications* has a lay term frequency of 35. Thus, *complication* is selected as the lay equivalent for *adverse effect*.

4 Results

We report on the two steps of our system: (1) finding semantic equivalents at the CUI level, and (2) finding an appropriate lay lexical equivalent.

Finding Semantic Equivalents at the CUI Level Table 5 shows the precision, recall and F-measure (computed as the harmonic mean between precision and recall) against our gold standard for the three alternative measures of association, including different combinations of these, and also adding the semantic filter. In addition, we report results for a competitive baseline based solely on CUI information, where *tech_term* and *lay_term* are considered equivalent if they have the same CUI.

The baseline is fairly competitive only because of its perfect precision (CUI in Table 5). Its recall, however (44.7), indicates that building a lexicon of technical and lay equivalents based solely on CUI information would miss too many pairs within the UMLS.

Method	P	R	F	Method	P	R	F	Method	P	R	F
lam	40.8	20.4	27.2	chi,odds	20.6	78.3	32.6	CUI	100	44.7	61.8
chi	38.7	23.7	29.4	chi,lam,odds	20.6	80.3	32.8	sem,odds	57.8	71.1	63.7
sem,lam	76.3	19.1	30.5	sem,chi	81.8	23.7	36.7	sem,lam,odds	57.4	73.7	64.6
odds	20.4	74.3	32	chi,lam	38.2	39.5	38.8	sem,chi,odds	58.5	75	65.7
lam,odds	20.5	77	32.3	sem,chi,lam	79.5	38.2	51.6	**sem,chi,lam,odds**	**57.9**	**77**	**66.1**

Table 5: Precision, Recall and F measures for different variants of the system.

Relying on only one measure of association without any semantic filtering to determine semantic equivalents is not a good strategy: λ* (lam in Table 5), χ^2(chi) and OR (odds), by themselves, yield the worst F measures. Interestingly, the measures of association identify different equivalent pairs in the pool of candidate pairs. Thus, combining them increases the coverage (or recall) of the system. For instance, λ* by itself has a low recall of 20.4 (lam). When combined with OR, it improves the recall from 74.3 (odds) to 77 (lam,odds); when combined with χ^2, it improves the recall from 23.7 (chi) to 39.5 (chi,lam). Combining the three measures of association (chi,lam,odds) yields the best recall (80.3), confirming our hypothesis that the measures are complementary and identify pairs with different characteristics in our corpus.

While combining measures of association improves recall, the semantic filter is very effective in filtering inaccurate pairs and, therefore, improving precision: λ*, for instance, improves from a precision of 40.8 (lam) to 76.3 (sem,lam) when the filter is added, with very little change in recall. The best variant of our system in terms of F measure is, not surprisingly, combining the three measures of association and adding the semantic filter (sem,chi,lam,odds in Table 5).

The results of these experiments are surprisingly good, considering that the contingency tables are built from a corpus of only 367 document pairs and rely on document frequency (not term frequency). These quantities are much smaller than those used in machine translation, for instance.

Finding Lay Lexical Equivalents We evaluate our strategy for finding an appropriate lay lexical item on the list of 152 term pairs identified by our medical expert as semantic equivalents. Our strategy achieves an accuracy of 86.7%.

5 Related Work

Our work belongs to the field of paraphrase identification. Much work has been done to build lexicons of semantically equivalent phrases. In generation systems, a lexicon is built manually (Robin, 1994) or by relying on an electronic thesaurus like WordNet (Langkilde and Knight, 1998) and setting constraints on the type of accepted paraphrases (for instance, accepting only synonyms as paraphrases, and not hypernyms). Building paraphrase lexicons from a corpus has also been investigated. Jacquemin and colleagues (1997) identify morphological and syntactic variants of technical terms. Barzilay and McKeown (2001) identify multi-word paraphrases from a sentence-aligned corpus of monolingual parallel texts. One interesting finding of this work is that the mined paraphrases were distributed across different semantic links in WordNet: some paraphrases had a hypernym relation, while others were synonyms, and others had no semantic links at all. The composition of our gold standard confirms this finding, since half of the semantically equivalent terms had different CUIs (see Table 3 for examples of such pairs).

If we consider technical and lay writing styles as two sublanguages, it is easy to see an analogy between our task and that of machine translation. Identifying translations for words or phrases has been deeply investigated in the field of statistical machine translation. The IBM models of word alignments are the basis for most algorithms to date. All of these are instances of the EM algorithm (Expectation Maximization) and rely on large corpora aligned at the sentence level. We cannot apply an EM-based model to our task since we have a very small corpus of paired technical/lay documents, and EM requires large amounts of data to achieve accurate results. Moreover, the technical and lay documents are not parallel, and thus, we do not have access to a sen-

tence alignment. Of course, our task is easier than the one of machine translation, since we focus on "translating" only technical terms, rather than every single word in a technical document.

Gale and Church (1991) do not follow the EM model, but rather find French translations of English words using a χ^2-like measure of association. Their corpus is the parallel, sentence-aligned Hansard corpus. Our method differs from theirs, as we do build the contingency table based on document frequencies. Gale and Church employ sentence-level frequencies. Our corpus is much smaller, and the sentences are not aligned (for comparison, we have 367 document-pairs, while they have nearly 900,000 sentence pairs). Another difference between our approach and theirs is our use of the semantic filter based on UMLS. We can afford to have such a filter because we focus on finding semantic equivalents of UMLS terms only.

6 Conclusions and Future Work

We presented an unsupervised method for identifying pairs of semantically equivalent technical/lay terms. Such a lexicon would benefit research in health literacy. In particular, it would benefit a system which automatically adapts a medical technical text to different levels of medical expertise.

We collected a corpus of pairs of technical/lay documents, where both documents convey similar information, but each is written for a different audience. Based on this corpus, we designed a method based on three alternative measures of association and a semantic filter derived from the UMLS. Our experiments show that combining data-driven statistics and a knowledge-based filter provides the best results.

Our method is concerned specifically with pairs of terms, as recognized from UMLS. While UMLS provides high coverage for technical terms, that is not the case for lay terms. In the future, we would like to extend our investigation to pairs consisting of a technical term and any noun phrase which is sufficiently frequent in our lay collection. Finding such pairs would have the side effect of augmenting UMLS, a primarily technical resource, with mined lay terms. One probable step towards this goal will be to increase the size of our corpus of paired technical and lay documents.

References

R. Barzilay and K. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proc. ACL'01*, pages 50–57.

N. Elhadad. 2006. Comprehending technical texts: Predicting and defining unfamiliar terms. In *Proc. AMIA'06*, pages 239–243.

J. Fleiss, B. Levin, and M.C. Paik. 2003. *Statistical Methods for Rates and Proportions*. Wiley.

W. Gale and K. Church. 1991. Identifying word correspondences in parallel texts. In *Proc. Speech and Natural Language Workshop*, pages 152–157.

L. Goodman and W. Kruskal. 1979. *Measures of Association for Cross Classifications*. Springler Verlag.

C. Jacquemin, J. Klavans, and E. Tzoukermann. 1997. Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In *Proc. ACL'97*, pages 24–31.

I. Langkilde and K. Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proc. COLING-ACL'98*, pages 704–710.

C. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

A. McCray. 2005. Promoting health literacy. *JAMA*, 12(2):152–163.

A. McEnery and Z. Xiao. 2007. Parallel and comparable corpora: What is happening? In *Incorporating Corpora. The Linguist and the Translator*. Clevedon.

National Library of Medicine, Bethesda, Maryland, 1995. *Unified Medical Language System (UMLS) Knowledge Sources*. `http://www.nlm.nih.gov/research/umls/`.

J. Robin. 1994. *Revision-Based Generation of Natural Language Summaries Providing Historical Background*. Ph.D. thesis, Columbia University.

R. Rudd, B. Moeykens, and T. Colton. 1999. *Annual Review of Adult Learning and Literacy*, chapter 5. Health and literacy: a review of medical and public health literature. Jossey Bass.

S. Teufel and N. Elhadad. 2002. Collection and Linguistic Processing of a Large-scale Corpus of Medical Articles. In *Proc. LREC'02*, pages 1214–1218.

Q. Zeng, E. Kim, J. Crowell, and T. Tse. 2005. A text corpora-based estimation of the familiarity of health terminology. In *Proc. ISBMDA'05*, pages 184–192.

Annotation of Chemical Named Entities

Peter Corbett
Cambridge University
Chemical Laboratory
Lensfield Road
Cambridge
UK CB2 1EW
ptc24@cam.ac.uk

Colin Batchelor
Royal Society of Chemistry
Thomas Graham House
Milton Road
Cambridge
UK CB4 0WF
batchelorc@rsc.org

Simone Teufel
Natural Language and
Information Processing Group
Computer Laboratory
University of Cambridge
UK CB3 0FD
sht25@cam.ac.uk

Abstract

We describe the annotation of chemical named entities in scientific text. A set of annotation guidelines defines 5 types of named entities, and provides instructions for the resolution of special cases. A corpus of full-text chemistry papers was annotated, with an inter-annotator agreement F score of 93%. An investigation of named entity recognition using LingPipe suggests that F scores of 63% are possible without customisation, and scores of 74% are possible with the addition of custom tokenisation and the use of dictionaries.

1 Introduction

Recent efforts in applying natural language processing to natural science texts have focused on the recognition of genes and proteins in biomedical text. These large biomolecules are—mostly—conveniently described as sequences of subunits, strings written in alphabets of 4 or 20 letters. Advances in sequencing techniques have lead to a boom in genomics and proteomics, with a concomitant need for natural language processing techniques to analyse the texts in which they are discussed.

However, proteins and nucleic acids provide only a part of the biochemical picture. Smaller chemical species, which are better described atom-by-atom, play their roles too, both in terms of their interactions with large biomolecules like proteins, and in the more general biomedical context. A number of resources exist to provide chemical information to the biological community. For example, the National Center For Biotechnology Information (NCBI) has added the chemical database PubChem[1] to its collections of bioinformatics data, and the ontology ChEBI (Chemical Entities of Biological Interest) (de Matos et al., 2006) has been added to the Open Biological Ontologies (OBO) family.

Small-molecule chemistry also plays a role in biomedical natural language processing. PubMed has included abstracts from medicinal chemistry journals for a long time, and is increasingly carrying other chemistry journals too. Both the GENIA corpus (Kim et al., 2003) and the BioIE cytochrome P450 corpus (Kulick et al., 2004) come with named entity annotations that include a proportion of chemicals, and at least a few abstracts that are recognisable as chemistry abstracts.

Chemical named entity recognition enables a number of applications. Linking chemical names to chemical structures, by a mixture of database lookup and the parsing of systematic nomenclature, allows the creation of semantically enhanced articles, with benefits for readers. An example of this is shown in the Project Prospect[2] annotations by the Royal Society of Chemistry (RSC). Linking chemical NER to chemical information retrieval techniques allows corpora to be searched for chemicals with similar structures to a query molecule, or chemicals that contain a particular structural motif (Corbett and Murray-Rust, 2006). With information extraction techniques, chemicals could be linked to their properties, applications and reactions, and with traditional gene/protein NLP techniques, it could be pos-

[1] http://pubchem.ncbi.nlm.nih.gov/
[2] http://www.projectprospect.org/

BioNLP 2007: Biological, translational, and clinical language processing, pages 57–64,
Prague, June 2007.

sible to discover new links between chemical data and bioinformatics data.

A few chemical named entity recognition (Corbett and Murray-Rust, 2006; Townsend et al., 2005; Vasserman, 2004; Kemp and Lynch, 1998; Sun et al., 2007) or classification (Wilbur et al., 1999) systems have been published. A plugin for the GATE system[3] will also recognise a limited range of chemical entities. Other named entity recognition or classification systems (Narayanaswamy et al., 2003; Torii et al., 2004; Torii and Vijay-Shanker, 2002; Spasic and Ananiadou, 2004) sometimes include chemicals as well as genes, proteins and other biological entities. However, due to differences in corpora and the scope of the task, it is difficult to compare them. There has been no chemical equivalent of the JNLPBA (Kim et al., 2004) or BioCreAtIvE (Yeh et al., 2005) evaluations. Therefore, a corpus and a task definition are required.

To find an upper bound on the levels of performance that are available for the task, it is necessary to study the inter-annotator agreement for the manual annotation of the texts. In particular, it is useful to see to what extent the guidelines can be applied by those not involved in their development. Producing guidelines that enable a highly consistent annotation may raise the quality of the results of any machine-learning techniques that use training data applied to the guidelines, and producing guidelines that cover a broad range of subdomains is also important (Dingare et al., 2005).

2 Annotation Guidelines

We have prepared a set of guidelines for the annotation of the names of chemical compounds and related entities in scientific papers. These guidelines grew out of work on PubMed abstracts, and have since been developed with reference to organic chemistry journals, and later a range of journals encompassing the whole of chemistry.

Our annotation guidelines focus on the chemicals themselves; we believe that these represent the major source of rare words in chemistry papers, and are of the greatest interest to end-users. Furthermore, many chemical names are formed systematically or semi-systematically, and can be interpreted without resorting to dictionaries and databases. As well as chemical names themselves, we also consider other words or phrases that are formed from chemical names.

The various types are summarised in Table 1.

Type	Description	Example
CM	chemical compound	citric acid
RN	chemical reaction	1,3-dimethylation
CJ	chemical adjective	pyrazolic
ASE	enzyme	methylase
CPR	chemical prefix	1,3-

Table 1: Named entity types

The logic behind the classes is best explained with an example drawn from the corpus described in the next section:

> In addition, we have found in previous studies that the Zn^{2+}–Tris system is also capable of efficiently hydrolyzing other β-lactams, such as clavulanic acid, which is a typical mechanism-based inhibitor of active-site serine β-lactamases (clavulanic acid is also a fairly good substrate of the zinc-β-lactamase from *B. fragilis*).

Here, 'clavulanic acid' is a specific chemical compound (a CM), referred to by a trivial (unsystematic) name, and 'β-lactams' is a class of chemical compounds (also a CM), defined by a particular structural motif. 'Zn^{2+}–Tris' is another CM (a complex rather than a molecule), and despite being named in an *ad hoc* manner, the name is compositional and it is reasonably clear to a trained chemist what it is. 'Serine' (another CM) can be used to refer to an amino acid as a whole compound, but in this case refers to it as a part of a larger biomolecule. The word 'hydrolyzing' (an RN) denotes a reaction involving the chemical 'water'. 'β-lactamases' (an ASE) denotes a class of enzymes that process β-lactams, and 'zinc-β-lactamase' (another ASE) denotes a β-lactamase that uses zinc. By our guidelines, the terms 'mechanism-based inhibitor' or 'substrate' are not annotated, as they denote a chemical role, rather than giving information about the structure or composition of the chemicals.

[3] `http://www.gate.ac.uk/`

The full guidelines occupy 31 pages (including a quick reference section), and contain 93 rules. Almost all of these have examples, and many have several examples.

A few distinctions need to be explained here. The classes RN, CJ and ASE do not include all reactions, adjectives or enzymes, but only those that entail specific chemicals or classes of chemicals—usually by being formed by the modification of a chemical name—for example, 'β-lactamases' in the example above is formed from the name of a class of chemicals. Words derived from Greek and Latin words for 'water', such as 'aqueous' and 'hydrolysis', are included when making these annotations.

The class CPR consists of prefixes, more often found in systematic chemical names, giving details of the geometry of molecules, that are attached to normal English words. For example, the chemical 1,2-diiodopentane is a 1,2-disubstituted pentane, and the '1,2-' forms the CPR in '1,2-disubstituted'. Although these contructions sometimes occur as infixes within chemical names, we have only seen these used as prefixes outside of them. We believe that identifying these prefixes will be useful in the adaptation of lexicalised parsers to chemical text.

The annotation task includes a small amount of word sense disambiguation. Although most chemical names do not have non-chemical homonyms, a few do. Chemical elements, and element symbols, give particular problems. Examples of this include 'lead', 'In' (indium), 'As' (arsenic), 'Be' (beryllium), 'No' (nobelium) and 'K' (potassium—this is confusable with Kelvin). These are only annotated when they occur in their chemical sense.

2.1 Related Work

We know of two publicly available corpora that also include chemicals in their named-entity markup. In both of these, there are significant differences to many aspects of the annotation. In general, our guidelines tend to give more importance to concepts regarding chemical structure, and less importance to biological role, than the other corpora do.

The GENIA corpus (Kim et al., 2003) includes several different classes for chemicals. Our class CM roughly corresponds to the union of GENIA's `atom`, `inorganic`, `other_organic_compound`, `nucleotide` and `amino_acid_monomer` classes, and also parts of `lipid` and `carbohydrate` (we exclude macromolecules such as lipoproteins and lipopolysaccharides). Occasionally terms that match our class RN are included as `other_name`. Our CM class also includes chemical names that occur within enzyme or other protein names (e.g. 'inosine-5′-monophosphate' in 'inosine-5′-monophosphate dehydrogenase') whereas the GENIA corpus (which allows nesting) typically does not. The GENIA corpus also sometimes includes qualifiers in terms, giving 'intracellular calcium' where we would only annotate 'calcium', and also includes some role/application terms such as 'antioxidant' and 'reactive intermediate'.

The BioIE P450 corpus (Kulick et al., 2004), by contrast, includes chemicals, proteins and other substances such as foodstuffs in a single category called 'substance'. Again, role terms such as 'inhibitor' are included, and may be merged with chemical names to make entities such as 'fentanyl metabolites' (we would only mark up 'fentanyl'). Fragments of chemicals such as 'methyl group' are not marked up; in our annotations, the 'methyl' is marked up.

The BioIE corpus was produced with extensive guidelines; in the GENIA corpus, much more was left to the judgement of the annotators. These lead to inconsistencies, such as whether to annotate 'antioxidant' (our guidelines treat this as a biological role, and do not mark it up). We are unaware of an inter-annotator agreement study for either corpus.

Both of these corpora include other classes of named entities, and additional information such as sentence boundaries.

3 Inter-annotator Agreement

3.1 Related Work

We are unaware of any studies of inter-annotator agreement with regards to chemicals. However, a few studies of gene/protein inter-annotator agreement do exist. Demetriou and Gaizauskas (2003) report an F score of 89% between two domain experts for a task involving various aspects of protein science. Morgan *et al.* (2004) report an F score of 87% between a domain expert and a systems developer for *D. melanogaster* gene names. Vlachos and Gasperin (2006) produced a revised version of the

guidelines for the task, and were able to achieve an F score of 91%, and a kappa of 0.905, between a computational linguist and a domain expert.

3.2 Subjects

Three subjects took part in the study. Subject A was a chemist and the main author of the guidelines. Subject B was another chemist, highly involved in the development of the guidelines. Subject C was a PhD student with a chemistry degree. His involvement in the development of guidelines was limited to proof-reading an early version of the guidelines. C was trained by A, by being given half an hour's training, a test paper to annotate (which satisfied A that C understood the general principles of the guidelines), and a short debriefing session before being given the papers to annotate.

3.3 Materials

The study was performed on 14 papers (full papers and communications only, not review articles or other secondary publications) published by the Royal Society of Chemistry. These were taken from the journal issues from January 2004 (excluding a themed issue of one of the journals). One paper was randomly selected to represent each of the 14 journals that carried suitable papers. These 14 papers represent a diverse sample of topics, covering areas of organic, inorganic, physical, analytical and computational chemistry, and also areas where chemistry overlaps with biology, environmental science, materials and mineral science, and education.

From these papers, we collected the title, section headings, abstract and paragraphs, and discarded the rest. To maximise the value of annotator effort, we also automatically discarded the experimental sections, by looking for headers such as 'Experimental'. This policy can be justified thus: In chemistry papers, a section titled "Results and Discussion" carries enough information about the experiments performed to follow the argument of the paper, whereas the experimental section carries precise details of the protocols that are usually only of interest to people intending to replicate or adapt the experiments performed. It is increasingly common for chemistry papers not to contain an experimental section in the paper proper, but to include one in the supporting online information. Furthermore, experimental sections are often quite long and tedious to annotate, and previous studies have shown that named-entity recognition is easier on experimental sections too (Townsend et al., 2005).

A few experimental sections (or parts thereof) were not automatically detected, and instead were removed by hand.

3.4 Procedure

The papers were hand-annotated using our in-house annotation software. This software displays the text so as to preserve aspects of the style of the text such as subscripts and superscripts, and allows the annotators to freely select spans of text with character-level precision—the text was not tokenised prior to annotation. Spans were not allowed to overlap or to nest. Each selected span was assigned to exactly one of the five available classes.

During annotation the subjects were allowed to refer to the guidelines (explained in the previous section), to reference sources such as PubChem and Wikipedia, and to use their domain knowledge as chemists. They were not allowed to confer with anyone over the annotation, nor to refer to texts annotated during development of the guidelines. The training of subject C by A was completed prior to A annotating the papers involved in the exercise.

3.5 Evaluation Methodology

Inter-annotator agreement was measured pairwise, using the F score. To calculate this, all of the exact matches were found and counted, and all of the entities annotated by one annotator but not the other (and vice versa) were counted. For an exact match, the left boundary, right boundary and type of the annotation had to match entirely. Thus, if one annotator had annotated 'hexane–ethyl acetate' as a single entity, and the other had annotated it as 'hexane' and 'ethyl acetate', then that would count as three cases of disagreement and no cases of agreement. We use the F score as it is a standard measure in the domain—however, as a measure it has weaknesses which will be discussed in the next subsection.

Given the character-level nature of the annotation task, and that the papers were not tokenised, the task cannot sensibly be cast as a classification problem, and so we have not calculated any kappa scores.

Overall results were calculated using two methods. The first method was to calculate the total levels of agreement and disagreement across the whole corpus, and to calculate a total F score based on that. The second method was to calculate F scores for individual papers (removing a single paper that contained two named entities—neither of which were spotted by subject B—as an outlier), and to calculate an unweighted mean, standard deviation and 95% confidence intervals based on those scores.

3.6 Results and Discussion

Subjects	F (corpus)	F (average)	std. dev.
A–B	92.8%	92.9%±3.4%	6.2%
A–C	90.0%	91.4%±3.1%	5.7%
B–C	86.1%	87.6%±3.1%	5.7%

Table 2: Inter-annotator agreement results. ± values are 95% confidence intervals.

The results of the analysis are shown in Table 2. The whole-corpus F scores suggest that high levels of agreement (93%) are possible. This is equivalent to or better than quoted values for biomedical inter-annotator agreement. However, the poorer agreements involving C would suggest that some of this is due to some extra information being communicated during the development of the guidelines.

A closer analysis shows that this is not the case. A single paper, containing a large number of entities, is notable as a major source of disagreement between A and C, and B and C, but not A and B. Looking at the annotations themselves, the paper contained many repetitions of the difficult entity 'Zn^{2+}–Tris', and also of similar entities. If the offending paper is removed from consideration, the agreement between A and C exceeds the agreement between A and B.

This analysis is confirmed using the per-paper F scores. Two-tailed, pairwise t-tests (excluding the outlier paper) showed that the difference in mean F scores between the A–B and A–C agreements was not statistically significant at the 0.05 significance level; however, the differences between B–C and A–B, and between B–C and A–C were.

A breakdown of the inter-annotator agreements by type is shown in Table 3. CM and RN, at least, seem to be reliably annotated. The other classes are less easy to assess, due to their rarity, both in terms of their total occurrence in the corpus and the number of papers that contain them.

Type	F	Number
CM	93%	2751
RN	94%	79
CJ	56%	20
ASE	96%	25
CPR	77%	10

Table 3: Inter-annotator agreement, by type. F scores are corpus totals, between Subjects A and C. The number is the number of entities of that class found by Subject A.

We speculate that the poorer B–C agreement may be due to differing error rates in the annotation. In many cases, it was clear from the corpus that errors were made due to failing to spot relevant entities, or by failing to look up difficult cases in the guidelines. Although it is not possible to make a formal analysis of this, we suspect that A made fewer errors, due to a greater familiarity with the task and the guidelines. This is supported by the results, as more errors would be involved in the B–C comparison than in comparisons involving A, leading to higher levels of disagreement.

We have also examined the types of disagreements made. There were very few cases where two annotators had annotated an entity with the same start and end point, but a different type; there were 2 cases of this between A and C, and 3 cases in each of the other two comparisons. All of these were confusions between CM and CJ.

In the A–B comparison, there were 415 entities that were annotated by either A or B that did not have a corresponding exact match. 183 (44%) of those were simple cases where the two annotators did not agree as to whether the entity should be marked up or not (i.e. the other annotator had not placed any entity wholly or partially within that span). For example, some annotators failed to spot instances of 'water', or disagreed over whether 'fat' (as a synonym for 'lipid') was to be marked up.

The remainder of those disagreements are due to disagreements of class, of where the boundaries should be, of how many entities there should be in a given span, and combinations of the above. In all

of these cases, the fact that the annotators produce at least one entity each for a given case means that disagreements of this type are penalised harshly, and therefore are given disproportionate weight. However, it is also likely that disagreements over whether to mark an entity up are more likely to represent a simple mistake than a disagreement over how to interpret the guidelines; it is easy to miss an entity that should be marked up when scanning the text.

A particularly interesting class of disagreement concerns whether a span of text should be annotated as one entity or two. For example, 'Zn^{2+}–Tris' could be marked up as a single entity, or as 'Zn^{2+}' and 'Tris'. We looked for cases where one annotator had a single entity, the left edge of which corresponded to the left edge of an entity annotated by the other annotator, and the right edge corresponded to the right edge of a different entity. We found 43 cases of this. As in each of these cases, at least three entities are involved, this pattern accounts for at least 30% of the inter-annotator disagreement. Only 17 of these cases contained whitespace—in the rest of the cases, hyphens, dashes or slashes were involved.

4 Analysis of the Corpus

To generate a larger corpus, a further two batches of papers were selected and preprocessed in the manner described for the inter-annotator agreement study and annotated by Subject A. These were combined with the annotations made by Subject A during the agreement study, to produce a corpus of 42 papers.

Type	Entities		Papers	
CM	6865	94.1%	42	100%
RN	288	4.0%	23	55%
CJ	60	0.8%	20	48%
ASE	31	0.4%	5	12%
CPR	53	0.7%	9	21%

Table 4: Occurrence of entities in the corpus, and numbers of papers containing at least one entity of a type.

From Table 4 it is clear that CM is by far the most common type of named entity in the corpus. Observation of the corpus shows that RN is common in certain genres of paper (for example organic synthesis papers), and generally absent from other genres. ASE, too, is a specialised category, and did not occur much in this corpus.

A closer examination of CM showed more than 90% of these to contain no whitespace. However, this is not to say that there are not significant numbers of multi-token entities. The difficulty of tokenising the corpus is illustrated by the fact that 1114 CM entities contained hyphens or dashes, and 388 CM entities were adjacent to hyphens or dashes in the corpus. This means that any named entity recogniser will have to have a specialised tokeniser, or be good at handling multi-token entities.

Tokenising the CM entities on whitespace and normalising their case revealed 1579 distinct words—of these, 1364 only occurred in one paper. There were 4301 occurrences of these words (out of a total of 7626). Whereas the difficulties found in gene/protein NER with complex multiword entities and polysemous words are less likely to be a problem here, the problems with tokenisation and large numbers of unknown words remain just as pressing.

As with biomedical text (Yeh et al., 2005), cases of conjunctive and disjunctive nomenclature, such as 'benzoic and thiophenic acids' and 'bromo- or chlorobenzene' exist in the corpus. However, these only accounted for 27 CM entities.

5 Named-Entity Recognition

To establish some baseline measures of performance, we applied the named-entity modules from the toolkit LingPipe,[4] which has been successfully applied to NER of *D. melanogaster* genes (e.g. by Vlachos and Gasperin (2006)). LingPipe uses a first-order HMM, using an enriched tagset that marks not only the positions of the named entities, but the tokens in front of and behind them. Two different strategies are employed for handling unknown tokens. The first (the `TokenShapeChunker`) replaces unknown or rare tokens with a morphologically-based classification. The second, newer module (the `CharLmHmmChunker`) estimates the probability of an observed word given a tag using language models based on character-level n-grams. The LingPipe developers suggest that the `TokenShapeChunker` typically outperforms the

[4] `http://www.alias-i.com/lingpipe/`

`CharLmHmmChunker`. However, the more sophisticated handling of unknown words by the `CharLmHmmChunker` suggests that it might be a good fit to the domain.

As well as examining the performance of LingPipe out of the box, we were also able to make some customisations. We have a custom tokeniser, containing several adaptations to chemical text. For example, our tokeniser will only remove brackets from the front and back of tokens, and only if that would not cause the brackets within the token to become unbalanced. For example, no brackets would be removed from '(*R*)-acetoin'. Likewise, it will only tokenise on a hyphen if the hyphen is surrounded by two lower-case letters on either side (and if the letters to the left are not common prehyphen components of chemical names), or if the string to the right has been seen in the training data to be hyphenated with a chemical name (e.g. 'derived' in 'benzene-derived'). By contrast, the default LingPipe tokeniser is much more aggressive, and will tokenise on hyphens and brackets wherever they occur.

The `CharLmHmmChunker`'s language models can also be fed dictionaries as additional training data—we have experimented with using a list of chemical names derived from ChEBI (de Matos et al., 2006), and a list of chemical elements. We have also made an extension to LingPipe's token classifier, which adds classification based on chemically-relevant suffixes (e.g. -yl, -ate, -ic, -ase, -lysis), and membership in the aforementioned chemical lists, or in a standard English dictionary.

We analysed the performance of the different LingPipe configurations by 3-fold cross-validation, using the 42-paper corpus described in the previous section. In each fold, 28 whole papers were used as training data, holding out the other 14 as test data. The results are shown in Table 5.

From Table 5, we can see that the character n-gram language models offer clear advantages over the older techniques, especially when coupled to a custom tokeniser (which gives a boost to F of over 7%), and trained with additional chemical names. The usefulness of character-based n-grams has also been demonstrated elsewhere (Wilbur et al., 1999; Vasserman, 2004; Townsend et al., 2005). Their use here in an HMM is particularly apt, as it allows the token-internal features in the language model to be combined with the token context.

Configuration	P	R	F
`TokenShape`	67.0%	52.9%	59.1%
+ c	71.2%	62.3%	66.5%
+ t	67.4%	52.5%	59.0%
+ c + t	73.3%	62.5%	67.4%
`CharLm`	62.7%	63.4%	63.1%
+ l	59.8%	68.8%	64.0%
+ t	71.1%	70.0%	70.5%
+ l + t	75.3%	73.5%	74.4%

Table 5: LingPipe performance using different configurations. c = custom token classifier, l = chemical name lists, t = custom tokeniser

The impact of custom tokenisation upon the older `TokenShapeChunker` is less dramatic. It is possible that tokens that contain hyphens, brackets and other special characters are more likely to be unknown or rare tokens—the `TokenShapeChunker` has previously been reported to make most of its mistakes on these (Vlachos and Gasperin, 2006), so tokenising them is likely to make less of an impact. It is also possible that chemical names are more distinctive as a string of subtokens rather than as one large token—this may offset the loss in accuracy from getting the start and end positions wrong. The `CharLmHmmChunker` already has a mechanism for spotting distinctive substrings such as 'N,N'-' and '-3-', and so the case for having long, well-formed tokens becomes much less equivocal.

It is also notable that improvements in tokenisation are synergistic with other improvements—the advantage of using the `CharLmHmmChunker` is much more apparent when the custom tokeniser is used, as is the advantage of using word lists as additional training data. It is notable that for the unmodified `TokenShapeChunker`, using the custom tokeniser actually harms performance.

6 Conclusion

We have produced annotation guidelines that enable the annotation of chemicals and related entities in scientific texts in a highly consistent manner. We have annotated a corpus using these guidelines, an analysis of which, and the results of using an off-

the-shelf NER toolkit, show that finding good approaches to tokenisation and the handling of unknown words is critical in the recognition of these entities. The corpus and guidelines are available by contacting the first author.

7 Acknowledgements

We thank Ann Copestake and Peter Murray-Rust for supervision, Andreas Vlachos and Advaith Siddharthan for valuable discussions, and David Jessop for annotation. We thank the RSC for providing the papers, and the UK eScience Programme and EPSRC (EP/C010035/1) for funding.

References

Peter T. Corbett and Peter Murray-Rust. 2006. High-Throughput Identification of Chemistry in Life Science Texts. *CompLife*, LNBI 4216:107–118.

P. de Matos, M. Ennis, M. Darsow, M. Guedj, K. Degtyarenko and R. Apweiler. 2006. ChEBI —Chemical Entities of Biological Interest. *Nucleic Acids Res*, Database Summary Paper 646.

George Demetriou and Rob Gaizauskas. 2003. Corpus resources for development and evaluation of a biological text mining system. *Proceedings of the Third Meeting of the Special Interest Group on Text Mining*, Brisbane, Australia, July.

Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning and Claire Grover. 2005. A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations. *Comparative and Functional Genomics*, 6(1-2),77-85.

Nick Kemp and Michael Lynch. 1998. Extraction of Information from the Text of Chemical Patents. 1. Identification of Specific Chemical Names. *J. Chem. Inf. Comput. Sci.*, 38:544-551.

J.-D. Kim, T. Ohta, Y. Tateisi and J. Tsujii. 2003. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180-i182.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi and Nigel Collier. 2004. Introduction to the Bio-Entity Recognition Task at JNLPBA. *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, 70-75.

Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein and Lyle Ungar. 2004. Integrated Annotation for Biomedical Information Extraction *HLT/NAACL BioLINK workshop*, 61-68.

Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37(6):396-410.

Meenakshi Narayanaswamy, K. E. Ravikumar and K. Vijay-Shanker. 2003. A Biological Named Entity Recogniser. *Pac. Symp. Biocomput.*, 427-438.

Irena Spasic and Sophia Ananiadou. 2004. Using Automatically Learnt Verb Selectional Preferences for Classification of Biomedical Terms. *Journal of Biomedical Informatics*, 37(6):483-497.

Bingjun Sun, Qingzhao Tan, Prasenjit Mitra and C. Lee Giles. 2007. Extraction and Search of Chemical Formulae in Text Documents on the Web. *The 16th International World Wide Web Conference (WWW'07)*, 251-259.

Manabu Torii and K. Vijay-Shanker. 2002. Using Unlabeled MEDLINE Abstracts for Biological Named Entity Classification. *Genome Informatics*, 13:567-568.

Manabu Torii, Sachin Kamboj and K. Vijay-Shanker. 2004. Using name-internal and contextual features to classify biological terms. *Journal of Biomedical Informatics*, 37:498-511.

Joe A. Townsend, Ann A. Copestake, Peter Murray-Rust, Simone H. Teufel and Christopher A. Waudby. 2005. Language Technology for Processing Chemistry Publications. *Proceedings of the fourth UK e-Science All Hands Meeting*, 247-253.

Alexander Vasserman. 2004. Identifying Chemical Names in Biomedical Text: An Investigation of the Substring Co-occurence Based Approaches. *Proceedings of the Student Research Workshop at HLT-NAACL*. 7-12.

Andreas Vlachos and Caroline Gasperin. 2006. Bootstrapping and Evaluating Named Entity Recognition in the Biomedical Domain. *Proceedings of BioNLP in HLT-NAACL*. 138-145.

W. John Wilbur, George F. Hazard, Jr., Guy Divita, James G. Mork, Alan R. Aronson and Allen C. Browne. 1999. Analysis of Biomedical Text for Chemical Names: A Comparison of Three Methods. *Proc. AMIA Symp.* 176-180.

Alexander Yeh, Alexander Morgan, Marc Colosimo and Lynette Hirschman. 2005. BioCreAtIvE Task IA: gene mention finding evaluation. *BMC Bioinformatics* 6(Suppl I):S2.

Recognising Nested Named Entities in Biomedical Text

Beatrice Alex, Barry Haddow and Claire Grover
School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh, EH8 9LW, UK
{balex,bhaddow,grover}@inf.ed.ac.uk

Abstract

Although recent named entity (NE) annotation efforts involve the markup of nested entities, there has been limited focus on recognising such nested structures. This paper introduces and compares three techniques for modelling and recognising nested entities by means of a conventional sequence tagger. The methods are tested and evaluated on two biomedical data sets that contain entity nesting. All methods yield an improvement over the baseline tagger that is only trained on flat annotation.

1 Introduction

Traditionally, named entity recognition (NER) has focussed on entities which are *continuous*, *non-nested* and *non-overlapping*. In other words, each token in the text belongs to at most one entity, and NEs consist of a continuous sequence of tokens. However, in some situations, it may make sense to relax these restrictions, for example by allowing entities to be *nested* inside other entities, or allowing *discontinuous* entities. GENIA (Ohta et al., 2002) and BioInfer (Pyysalo et al., 2007) are examples of recently produced NE-annotated biomedical corpora where entities nest. Corpora in other domains, for example the ACE[1] data, also contain nested entities.

This paper compares techniques for recognising nested entities in biomedical text. The difficulty of this task is that the standard method for converting NER to a sequence tagging problem with BIO-encoding (Ramshaw and Marcus, 1995), where each token is assigned a tag to indicate whether it is at the beginning (B), inside (I), or outside (O) of an entity, is not directly applicable when tokens belong to more than one entity. Here we explore methods of reducing the nested NER problem to one or more BIO problems so that existing NER tools can be used.

This paper is organised as follows. In Section 2, the problem of nested entities is introduced and motivated with examples from GENIA and our EPPI (enriched protein-protein interaction) data. Related work is reviewed in Section 3. The proposed techniques enabling NER for nested NEs are explained in Section 4. Section 5 details the experimental setup, including descriptive statistics of the corpora and specifics of the classifier. The results of comparing different tagging methods are analysed in Section 6, with a discussion and conclusion in Section 7.

2 Nested Entities

The majority of previous work on NER is conducted using data sets annotated either with continuous, non-nested and non-overlapping NEs or an annotation scheme reduced to a flat annotation of a similar kind in order to simplify the recognition task. However, annotated corpora often contain entities that are nested or discontinuous. For example, the GENIA corpus contains nested entities such as:

<RNA><DNA>CIITA</DNA> mRNA</RNA>

where the string "CIITA" denotes a DNA and the entire string "CIITA mRNA" refers to an RNA. Such nesting complicates the task of traditional NER systems, which generally rely on data represented with the BIO encoding or other flat annotation variations thereof. The majority of NER studies on corpora

[1] http://www.nist.gov/speech/tests/ace/index.htm

BioNLP 2007: Biological, translational, and clinical language processing, pages 65–72,
Prague, June 2007.

GENIA		EPPI	
Count	Nesting	Count	Nesting
3,614	(other_name (protein t) t)	1,698	(fusion (protein t) t (protein t))
907	(DNA (protein t) t)	1,269	(drug/compound (protein t))
856	(protein (protein t) t)	455	(fusion (fragment t) t (protein t))
661	(protein t (protein t))	412	(protein (protein t) t)
546	(other_name (DNA t) t)	361	(complex (protein t) t (protein t))
541	(other_name t (other_name t))	298	(fusion (protein t) t (fragment t))
470	(cell_type t (cell_type t))	246	(fragment t (fragment t))
351	(DNA t (DNA t))	241	(cell_line t (cell_line t))
326	(other_name (virus t) t)	207	(fragment (protein t))
262	(other_name (lipid t) t)	201	(fusion (protein t) t (mutant t))

Table 1: 10 most frequent types of nesting in the GENIA corpus and the combined TRAIN and DEVTEST sections of the EPPI data (see Section 5.1), where t represents the text.

containing nested structures focus on recognising the outermost (non-embedded) entities (e.g. Kim et al. 2004) , as they contain the most information, including that of embedded entities (Zhang et al., 2004). This enables a simplification of the recognition task to a sequential analysis problem.

Our aim is to recognise all levels of NE nesting occurring in two biomedical corpora: the GENIA corpus (Version 3.02) and the EPPI corpus (see Section 5.1). The latter data set was collected and annotated as part of the TXM project. Its annotation contains 9 different biomedical entities. While the GENIA corpus contains nested entities up to a level of four layers of embedding, the nested entities in the EPPI corpus only have three layers. Table 1 lists the ten most frequent types of entity nesting occurring in both corpora. In the remainder of the paper, we differentiate between:

embedded NEs: contained in other NEs

non-embedded NEs: not contained in other NEs

containing NEs: containing other NEs

non-containing NEs: not containing other NEs

The GENIA corpus is made up of a larger percentage of both embedded entity (18.61%) and containing entity (16.95%) mentions than the EPPI data (12.02% and 8.27%, respectively). In both corpora, nesting can occur in three different ways:

1. *Entities containing one or more shorter embedded entities.* Such nesting is very frequent in both data sets. For example, the DNA "IL-2 promoter" in the GENIA corpus contains the protein "IL-2". In the EPPI corpus, fusions and complexes often contain nested proteins, e.g. the complex "CBP/p300", where "CBP" and "p300" are marked as proteins.

2. *Entities with more than one entity type.* Although they occur in both data sets, they are very rare in the GENIA corpus. For example, the string "p21ras" is annotated both as DNA and protein. In the EPPI data, proteins can also be annotated as drug/compound, where it can be clearly established that the protein is used as a drug to affect the function of an organism, cell, or biological process.

3. *Coordinated entities.* Coordinated NEs account for approximately 2% of all NEs in the GENIA and EPPI data. In the original corpora they are annotated differently, but for this work they are all converted to a common format.[2] The outermost annotation of coordinated structures and any continuous entity mark-up within them is retained. For example, in "human interleukin-2 and -4" both the continuous embedded entity "human interleukin-2" and the entire string are marked as proteins. The markup for discontinuous embedded entities, like "human interleukin-4" in the previous example, is not retained, as they could be derived in a post-processing step once nested entities are recognised.

3 Related Work

In previous work addressing nested entities, Shen et al. (2003), Zhang et al. (2004), Zhou et al. (2004), Zhou (2006), and Gu (2006) considered the GENIA

[2]Both corpora are represented in XML with standoff annotation, potentionally allowing overlapping NEs.

corpus, where nested entities are relatively frequent. All these studies ignore embedded entities occurring in coordinated structures and only retain their outermost annotation. Shen et al. (2003), Zhang et al. (2004), and Zhou et al. (2004) all report on a rule-based approach to dealing with nested NEs in the GENIA corpus (Version 3.0) in combination with a Hidden Markov Model (HMM) that first recognises innermost NEs. They use four basic hand-crafted patterns and a combination thereof to generate nesting rules from the training data and thereby derive NEs containing the innermost NEs. The experimental setup of these studies differs slightly. While Shen et al. (2003) and Zhang et al. (2004) report results testing on 4% of the abstracts in the GENIA corpus, Zhou et al. (2004) report 10-fold cross-validation scores. Zhou (2006) applies the same rule-based method for dealing with nested entities to the output of a mutual information independence model (MIIM) combined with a support vector machine (SVM) plus sigmoid. His results are based on 5-fold cross-validation on the GENIA corpus (Version 3.0). In each of the studies, the rule-based approach to nested entities results in an improvement of between 3.0 and 3.5 points in $F1$ over the baseline model. However, as explicitly stated by Shen et al. (2003) and Zhang et al. (2004), this evaluation is limited to non-embedded (i.e. top-level and non-nested) entities. The highest overall $F1$-score reported for all entities in the GENIA corpus is 71.2 (Zhou, 2006), which again only appears to reflect the performance on non-embedded entities.

Zhang et al. (2004) also compare the rule-based method with HMM-based cascaded recognition that extends iteratively from the shortest to the longest entities. Their basic HMM model is combined with HMM models trained on transformed cascaded annotations. During training, embedded entity terms are replaced by their entity type as a way of unnesting the data. During testing, subsequent iterations rely on the tagging of the first recognition pass and are repeated until no more entities are recognised. However, this method only results in an improvement of 1.2 points in $F1$ over their basic classifier.

Gu (2006) reports results on recognising nested entities in the GENIA corpus (Version 3.02) when training an SVM-light binary classifier to recognise either proteins or DNA. Training with the outermost labelling yields better performance on recognising outermost entities and, conversely, using the inner labelling results in highest scores for recognising inner entities. The best exact match $F1$-scores of 73.0 and 47.5 for proteins and DNA, respectively, are obtained when training on data with inner labelling and evaluating on the inner entities.

McDonald et al. (2005) propose structured multilabel classification as opposed to sequential labelling for dealing with nested, discontinuous, and overlapping NEs. This approach uses a novel text segment representation in preference to the BIO-encoding. Their corpus contains MEDLINE abstracts on the inhibition of the enzyme CYP450 (Kulick et al., 2004), specifically those abstracts that contain at least one overlapping and one discontinuous annotation. While this data does not contain nested NEs, discontinuous and overlapping NEs make up 6% of all NEs. The classifier performs competitively with sequential tagging models on continuous and non-overlapping entities for NER and noun phrase chunking. On discontinuous and overlapping NEs in the biomedical data alone, its best performance is 56.25 $F1$. As the corpus does not contain nested NEs, it would be of interest to investigate the algorithm's performance on the GENIA corpus.

4 Modelling Techniques

As large amounts of time and effort have been devoted to work on non-nested NER using the BIO-encoding approach, it would be useful if this work could be easily applied to nested NER. In this paper, three different ways of addressing nested NER will be compared: *layering*, *cascading*, and *joined label tagging*. All techniques aim to reduce the nested NER problem to one or more BIO problems, so that existing NER tools can be used. Table 2 shows an example representation for each modelling technique of the following two non-nested and nested entity annotations found in a GENIA abstract:

<multi_cell>mice</multi_cell> ...
<other_name><RNA><protein>tumor necrosis factor-alpha</protein>
(<protein>TNF- alpha</protein>) messenger RNA</RNA> levels</other_name>

In layering, each level of nesting is modelled as a separate BIO problem. The output of models trained on individual layers is combined subsequent to tagging by taking the union. Layers can be created

Token	Inside-out layering			Outside-in layering		
Model	Layer 1	Layer 2	Layer 3	Layer 3	Layer 2	Layer 1
mice	B-multi_cell	O	O	B-multi_cell	O	O
…	…	…	…	…	…	…
tumor	B-protein	B-RNA	B-other_name	B-other_name	B-RNA	B-protein
necrosis	I-protein	I-RNA	I-other_name	I-other_name	I-RNA	I-protein
factor-alpha	I-protein	I-RNA	I-other_name	I-other_name	I-RNA	I-protein
(	O	I-RNA	I-other_name	I-other_name	I-RNA	O
TNF-alpha	B-protein	I-RNA	I-other_name	I-other_name	I-RNA	B-protein
)	O	I-RNA	I-other_name	I-other_name	I-RNA	O
messenger	O	I-RNA	I-other_name	I-other_name	I-RNA	O
RNA	O	I-RNA	I-other_name	I-other_name	I-RNA	O
levels	O	O	I-other_name	I-other_name	O	O

	Cascading			Joined label tagging
Model	All entity types	other	RNA	Joined labels
mice	B-multi_cell	O	O	B-multi_cell+O+O
…	…	…	…	…
tumor	B-protein	B-other_name	B-RNA	B-protein+B-RNA+B-other_name
necrosis	I-protein	I-other_name	I-RNA	I-protein+I-RNA+I-other_name
factor-alpha	I-protein	I-other_name	I-RNA	I-protein+I-RNA+I-other_name
(	O	I-other_name	I-RNA	O+I-RNA+I-other_name
TNF-alpha	B-protein	I-other_name	I-RNA	B-protein+I-RNA+I-other_name
)	O	I-other_name	I-RNA	O+I-RNA+I-other_name
messenger	O	I-other_name	I-RNA	O+I-RNA+I-other_name
RNA	O	I-other_name	I-RNA	O+I-RNA+I-other_name
levels	O	I-other_name	O	O+O+I-other_name

Table 2: Example representation of nested entities for various modelling techniques.

inside-out or *outside-in*. For inside-out layering, the first layer is made up of all non-containing entities, the second layer is composed of all those entities which only contain one layer of nesting, etc. Conversely, outside-in layering means that the first layer contains all non-embedded entities, the second layer contains all entities which are only contained within one outer entity, etc. Both directions of layering can be modelled using a conventional NE tagger.

Cascading reduces the nested NER task to several BIO problems by grouping one or more entity types and training a separate model for each group. Again, the output from individual models is combined during tagging. Subsequent models in the cascade may have access to the guesses of previous ones by means of a GUESS feature. The cascaded method is unable to recognise entities containing entities of the same type, which may be a drawback for some data sets. Cascading also raises the issue of how to group entity types. This is dependent on the types of entities that nest within a given data set and would potentially require large amounts of experimentation to determine the best combination. Moreover, training a model for each entity type lengthens training time considerably, and may degrade performance due to the dominance of the O tags for infrequent categories. It is possible, however, to create a cascaded tagger combining one model trained on all entity types with models trained on entity types that frequently contain other entities.

Finally, joined label tagging entails creating one tagging problem for all entities by concatenating the BIO tags of all levels of nesting. A conventional named entity recogniser is then trained on the data containing the joined labels. Once the classifier has assigned the joined labels during tagging, they are decoded into their original BIO format for each individual entity type. Compared to the other techniques, joined label tagging involves a much larger tag set, which can increase dramatically with the number of entity types occurring in a data set. This can result in data sparsity which may have a detrimental effect on performance.

5 Experimental Setup

5.1 Corpora

GENIA (V3.02), a large publicly available biomedical corpus annotated with biomedical NEs, is widely used in the text mining community (Cohen et al., 2005). This data set consists of 2,000 MEDLINE abstracts in the domain of molecular biology ($\simeq$0.5m tokens). The annotations used for the experiments

reported here are based on the GENIA ontology, published in Ohta et al. (2002). It contains the following classes: amino acid monomer, atom, body part, carbohydrate, cell component, cell line, cell type, DNA, inorganic, lipid, mono-cell, multi-cell, nucleotide, other name, other artificial source, other organic compound, peptide, polynucleotide, protein, RNA, tissue, and virus. In this work, protein, DNA and RNA sub-types are collapsed to their super-type, as done in previous studies (e.g. Zhou 2006). To the best of our knowledge, no inter-annotator agreement (IAA) figures on the NE-annotation in the GENIA corpus are reported in the literature.

The EPPI corpus consists of 217 full-text papers selected from PubMed and PubMedCentral as containing protein-protein interactions (PPIs). The papers were either retrieved in XML or HTML, depending on availability, and converted to an internal XML format. Domain experts annotated all documents for NEs and PPIs, as well as extra (enriched) information associated with PPIs and normalisations of entities to publicly available ontologies. The entity annotations are the focus of the current work. The types of entities annotated in this data set are: complex, cell line, drug/compound, experimental method, fusion, fragment, modification, mutant, and protein. Out of the 217 papers, 125 were singly annotated, 65 were doubly annotated, and 27 were triply annotated. The IAA, measured by taking the $F1$ score of one annotator with respect to another when the same paper is annotated by two different annotators, ranges from 60.40 for the entity type mutant to 91.59 for protein, with an overall micro-averaged $F1$-score of 84.87. The EPPI corpus ($\backsimeq$2m tokens) is divided into three sections, TRAIN (66%), DEVTEST (17%), and TEST (17%), with TEST only to be used for final evaluation, and not to be consulted by the researchers in the development and feature optimisation phrase. The experiments described here involve the EPPI TRAIN and DEVTEST sets.

5.2 Pre-processing

All documents are passed through a sequence of pre-processing steps implemented using the LT-XML2 and LT-TTT2 tools (Grover et al., 2006) with the output of each step encoded in XML mark-up. Tokenisation and sentence splitting is followed by part-of-speech tagging with the Maximum Entropy Markov Model (MEMM) tagger developed by Curran and Clark (2003) (hereafter referred to as C&C) for the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003), trained on the MedPost data (Smith et al., 2004). Information on lemmatisation, as well as abbreviations and their long forms, is added using the *morpha* lemmatiser (Minnen et al., 2000) and the *ExtractAbbrev* script of Schwartz and Hearst (2003), respectively. A lookup step uses ontological information to identify scientific and common English names of species. Finally, a rule-based chunker marks up noun and verb groups and their heads (Grover and Tobin, 2006).

5.3 Named Entity Tagging

The C&C tagger, referred to earlier, forms the basis of the NER component of the TXM natural language processing (NLP) pipeline designed to detect entity relations and normalisations (Grover et al., 2007). The tagger, in common with many ML approaches to NER, reduces the entity recognition problem to a sequence tagging problem by using the BIO encoding of entities. As well as performing well on the CoNLL-2003 task, Maximum Entropy Markov Models have also been successful on biomedical NER tasks (Finkel et al., 2005). As the vanilla C&C tagger (Curran and Clark, 2003) is optimised for performance on newswire text, various modifications were applied to improve its performance for biomedical NER. Table 3 lists the extra features specifically designed for biomedical text. The C&C tagger was also extended using several gazetteers, including a protein, complex, experimental method and modification gazetteer, targeted at recognising entities occurring in the EPPI data. Further post-processing specific to the EPPI data involves correcting boundaries of some hyphenated proteins and filtering out entities ending in punctuation.

All experiments with the C&C tagger involve 5-fold cross-validation on all 2,000 GENIA abstracts and the combined EPPI TRAIN and DEVTEST sets. Cross-validation is carried out at the document level. For simple tagging, the C&C tagger is trained on the non-containing entities (innermost) or on the non-embedded entities (outermost). For inside-out and outside-in layering, a separate C&C model is trained for each layer of entities in the data, i.e. four models for the GENIA data and three models for the EPPI data. Cascading is performed on individual entities with different orderings, either ordering en-

Feature	Description
CHARACTER	Regular expressions matching typical protein names
WORDSHAPE	Extended version of the C&C WORDTYPE feature
HEADWORD	Head word of the current noun phrase
ABBREVIATION	Term identified as an abbreviation of a gazetteer term within a document
TITLE	Term seen in a noun phrase in the document title
WORDCOUNTER	Non-stop word that is among the 10 most frequent ones in a document
VERB	Verb lemma information added to each noun phrase token in the sentence
FONT	Text in italic and subscript contained in the original document format

Table 3: Extra features added to C&C .

tity models according to performance or entity frequency in the training data, ranging from highest to lowest. Cascading is also carried out on groups of entities (e.g. one model for all entities, one for a specific entity type, and combinations). Subsequent models in the cascade have access to the guesses of previous ones via a GUESS feature. Finally, joined label tagging is done by concatenating individual BIO tags from the innermost to the outermost layer.

As in the GENIA corpus, the most frequently annotated entity type in the EPPI data is protein with almost 55% of all annotations in the combined TRAIN and DEVTEST data (see Table 5). Given that the scores reported in this paper are calculated as $F1$ micro-averages over all categories, they are strongly influenced by the classifier's performance on proteins. However, scoring is not limited to a particular layer of entities (e.g. only outermost layer), but includes all levels of nesting. During scoring, a correct match is achieved when exactly the same sequence of text (encoded in start/end offsets) is marked with the same entity type in the gold standard and the system output. Precision, recall and $F1$ are calculated in standard fashion from the number of true positive, false positive and false negative NEs recognised.

6 Results

Table 4 lists overall cross-validation $F1$-scores calculated for all NEs at all levels of nesting when applying the various modelling techniques. For GENIA, cascading on individual entities when ordering entity models by performance yields the highest $F1$-score of 67.88. Using this method yields an increase of 3.26 $F1$ over the best simple tagging method, which scores 64.62 $F1$. Joined label tagging results in the second best overall $F1$-score of 67.82. Both layering (inside-out) and cascading (combining a model trained on all NEs with 4 models trained on other name, DNA, protein, or RNA) also perform competitively, reaching $F1$-scores of 67.62 and 67.56, respectively. In the experiments with the EPPI corpus, cascading is also the winner with an $F1$-score of 70.50 when combining a model trained on all NEswith one trained on fusions. This method only results in a small, yet statistically significant (χ^2, $p \leq 0.05$), increase in $F1$ of 0.43 over the best simple tagging algorithm. This could be due to the smaller number of nested NEs in the EPPI data and the fact that this data contains many NEs with more than one category. Layering (inside-out) performs almost as well as cascading ($F1$=70.44).

The difference in the overall performance between the GENIA and the EPPI corpus is partially due to the difference in the number of NEs which C&C is required to recognise, but also due to the fact that all features used are optimised for the EPPI data and simply applied to the GENIA corpus. The only feature not used for the experiments with the GENIA corpus is FONT, as this information is not preserved in the original XML of that corpus.

7 Discussion and Conclusion

According to the results for the modelling techniques, each proposed method outperforms simple tagging. Cascading yields the best result on the GENIA ($F1$=67.88) and EPPI data ($F1$=70.50), see Table 5 for individual entity scores. However, it involves extensive amounts of experimentation to determine the best model combination. The best setup for cascading is clearly data set dependent. With larger numbers of entity types annotated in a given corpus, it becomes increasingly impractical to exhaustively test all possible orders and combinations. Moreover, training and tagging times are lengthened as more models are combined in the cascade.

GENIA V3.02		EPPI	
Technique	$F1$	Technique	$F1$
Simple Tagging			
Training on innermost entities	64.62	Training on innermost entities	70.07
Training on outermost entities	62.72	Training on outermost entities	69.18
Layering			
Inside-out	*67.62*	Inside-out	*70.44*
Outside-in	*67.02*	Outside-in	70.21
Cascading			
Individual NE models (by performance)	***67.88***	Individual NE models (by performance)	70.42
Individual NE models (by frequency)	*67.72*	Individual NE models (by frequency)	*70.43*
All-cell_type	64.55	All-complex	70.03
All-DNA	*65.02*	All-drug/compound	70.08
All-other_name	*66.99*	All-fusion	***70.50***
All-protein	64.77	All-protein	70.02
All-RNA	64.80	All-complex-fusion	*70.46*
All-other_name-DNA-protein-RNA	*67.56*	All-drug/compound-fusion	*70.50*
Joined label tagging			
Inside-out	*67.82*	Inside-out	70.37

Table 4: Cross-validation $F1$-scores for different modelling techniques on the GENIA and EPPI data. Scores in italics mark statistically significant improvements (χ^2, $p \leq 0.05$) over the best simple tagging score.

Despite the large number of tags involved in using joined label tagging, this method outperforms simple tagging for both data sets and even results in the second-best overall $F1$-score of 67.72 obtained for the GENIA corpus. The fact that joined label tagging only requires training and tagging with one model makes this approach a viable alternative to cascading which is far more time-consuming to run.

Inside-out layering performs competitively both for the GENIA corpus ($F1$=67.62) and the EPPI corpus ($F1$=70.37), considering how little time is involved in setting up such experiments. As with joined label tagging, minimal optimisation is required when using this method. One disadvantage (as compared to simple, and to some extent joined label tagging) is that training and tagging times increase with the number of layers that are modelled.

In conclusion, this paper introduced and tested three different modelling techniques for recognising nested NEs, namely layering, cascading, and joined label tagging. As each of them reduces nested NER to one or more BIO-encoding problems, a conventional sequence tagger can be used. It was shown that each modelling technique outperfoms the simple tagging method for both biomedical data sets.

Future work will involve testing the proposed techniques on other data sets containing entity nesting, including the ACE data. We will also determine their merit when applying a different learning algorithm. Furthermore, possible solutions for recognising discontinuous entities will be investigated.

8 Acknowledgements

The authors are very grateful to the annotation team, and to Cognia (`http://www.cognia.com`) for their collaboration on the TXM project. This work is supported by the Text Mining Programme of ITI Life Sciences Scotland (`http://www.itilifesciences.com`).

References

K. Bretonnel Cohen, Lynne Fox, Philip V. Ogren, and Lawrence Hunter. 2005. Corpus design for biomedical natural language processing. In *Proceedings of the ACL-ISMB workshop on linking biological literature, ontologies and databases: mining biological semantics*, pages 38–45.

James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-2003*, pages 164–167.

GENIA V3.02					EPPI				
Entity type	Count	P	R	$F1$	Entity type	Count	P	R	$F1$
All	94,014	69.3	66.5	67.9	All	134,059	73.1	68.1	70.5
protein	34,813	75.1	74.9	75.0	protein	73,117	76.2	82.1	79.0
other name	20,914	60.0	67.2	63.4	expt. method	12,550	74.3	72.4	73.3
DNA	10,589	64.2	57.5	60.6	fragment	11,571	54.5	41.7	47.3
cell type	7,408	71.2	69.2	70.2	drug/compound	10,236	64.9	37.7	47.7
other org. compound	4,109	76.6	57.8	65.9	cell line	6,505	68.3	53.4	59.9
cell line	4,081	66.3	53.8	59.4	complex	6,454	62.5	32.2	42.5
lipid	2,359	76.9	65.6	70.8	modification	5,727	95.4	94.2	94.8
virus	2,133	76.0	73.4	74.7	mutant	4,025	40.7	23.2	29.6
multi-cell	1,784	72.5	60.1	65.7	fusion	3,874	56.6	36.0	44.0

Table 5: Individual counts and scores of the most frequent GENIA and all EPPI entity types for the best-performing method: cascading.

Jenny Rose Finkel, Shipra Dingare, Christopher D. Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. 2005. Exploring the boundaries: Gene and protein identification in biomedical text. *BMC Bioinformatics*, 6(Suppl1):S5.

Claire Grover and Richard Tobin. 2006. Rule-based chunking and reusability. In *Proceedings of LREC 2006*, pages 873–878.

Claire Grover, Michael Matthews, and Richard Tobin. 2006. Tools to address the interdependence between tokenisation and standoff annotation. In *Proceedings of NLPXML 2006*, pages 19–26.

Claire Grover, Barry Haddow, Ewan Klein, Michael Matthews, Leif Arda Nielsen, Richard Tobin, and Xinglong Wang. 2007. Adapting a relation extraction pipeline for the BioCreAtIvE II task. In *Proceedings of the BioCreAtIvE Workshop 2007*, Madrid, Spain.

Baohua Gu. 2006. Recognizing nested named entities in GENIA corpus. In *Proceedings of the BioNLP Worshop, HLT-NAACL 2006*, pages 112–113.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of JNLPBA 2004*, pages 70–75.

Seth Kulick, Ann Bies, Mark Liberman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, and Lyle Ungar. 2004. Integrated annotation for biomedical information extraction. In *Proceedings of BioLINK 2004*, pages 61–68.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *Proceedings of HLT/EMNLP 2005*, pages 987–994.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of INLG 2000*, pages 201–208.

Tomoko Ohta, Yuka Tateisi, Hideki Mima, and Jun'ichi Tsujii. 2002. GENIA corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of HLT 2002*, pages 73–77.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the 3rd Workshop on Very Large Corpora (ACL 1995)*, pages 82–94.

Ariel S. Schwartz and Marti A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing*, pages 451–462.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the BioNLP Workshop, ACL 2003*, pages 49–56.

Larry Smith, Tom Rindflesch, and W. John Wilbur. 2004. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *Journal of Biomedical Informatics*, 37(6):411–422.

Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and ChewLim Tan. 2004. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190.

Guodong Zhou. 2006. Recognizing names in biomedical texts using mutual information independence model and svm plus sigmoid. *International Journal of Medical Informatics*, 75:456–467.

Exploring the Efficacy of Caption Search for Bioscience Journal Search Interfaces

Marti A. Hearst, Anna Divoli, Jerry Ye
School of Information, UC Berkeley
Berkeley, CA 94720
{hearst,divoli,jerryye}@ischool.berkeley.edu

Michael A. Wooldridge
California Digital Library
Oakland, CA 94612
mikew@ucop.edu

Abstract

This paper presents the results of a pilot usability study of a novel approach to search user interfaces for bioscience journal articles. The main idea is to support search over figure captions explicitly, and show the corresponding figures directly within the search results. Participants in a pilot study expressed surprise at the idea, noting that they had never thought of search in this way. They also reported strong positive reactions to the idea: 7 out of 8 said they would use a search system with this kind of feature, suggesting that this is a promising idea for journal article search.

1 Introduction

For at least two decades, the standard way to search for bioscience journal articles has been to use the National Library of Medicine's PubMed system to search the MEDLINE collection of journal articles. PubMed has innovated search in many ways, but to date search in PubMed is restricted to the title, abstract, and several kinds of metadata about the document, including authors, Medical Subject Heading (MeSH) labels, publication year, and so on.

On the Web, searching within the full text of documents has been standard for more than a decade, and much progress has been made on how to do this well. However, until recently, full text search of bioscience journal articles was not possible due to two major constraints: (1) the full text was not widely available online, and (2) publishers restrict researchers from downloading these articles in bulk.

Recently, online full text of bioscience journal articles has become ubiquitous, eliminating one barrier. The intellectual property restriction is under attack, and we are optimistic that it will be nearly entirely diffused in a few years. In the meantime, the PubMedCentral Open Access collection of journals provides an unrestricted resource for scientists to experiment with for providing full text search.[1]

Full text availability requires a re-thinking of how search should be done on bioscience journal articles. One opportunity is to do information extraction (text mining) to extract facts and relations from the *body* of the text, as well as from the title and abstract as done by much of the early text mining work. (The Biocreative competition includes tasks that allow for extraction within full text (Yeh et al., 2003; Hirschman et al., 2005).) The results of text extraction can then be exposed in search interfaces, as done in systems like iHOP (Hoffmann and Valencia, 2004) and ChiliBot (Chen and Sharp, 2004) (although both of these search only over abstracts).

Another issue is how to adjust search ranking algorithms when using full text journal articles. For example, there is evidence that ranking algorithms should consider which section of an article the query terms are found in, and assign different weights to different sections for different query types (Shah et al., 2003), as seen in the TREC 2006 Genomics Track (Hersh et al., 2006).

Recently Google Scholar has provided search

[1] The license terms for use for BioMed Central can be found at: http://www.biomedcentral.com/info/authors/license and the license for PubMedCentral can be found at: http://www.pubmedcentral.gov/about/openftlist.html

BioNLP 2007: Biological, translational, and clinical language processing, pages 73–80,
Prague, June 2007.

over the full text of journal articles from a wide range of fields, but with no special consideration for the needs of bioscience researchers[2]. Google Scholar's distinguishing characteristic is its ability to show the number of papers that cite a given article, and rank papers by this citation count. We believe this is an excellent starting point for full text search, and any future journal article search system should use citation count as a metric. Unfortunately, citation count requires access to the entire collection of articles; something that is currently only available to a search system that has entered into contracts with all of the journal publishers.

In this article, we focus on another new opportunity: the ability to search over figure captions and display the associated figures. This idea is based on the observation, noted by our own group as well as many others, that when reading bioscience articles, researchers tend to start by looking at the title, abstract, figures, and captions. Figure captions can be especially useful for locating information about experimental results. A prominent example of this was seen in the 2002 KDD competition, the goal of which was to find articles that contained experimental evidence for gene products, where the top-performing team focused its analysis on the figure captions (Yeh et al., 2003).

In the Biotext project, we are exploring how to incorporate figures and captions into journal article search explicitly, as part of a larger effort to provide high-quality article search interfaces. This paper reports on the results of a pilot study of the caption search idea. Participants found the idea novel, stimulating, and most expressed a desire to use a search interface that supports caption search and figure display.[3]

2 Related Work

2.1 Automated Caption Analysis

Several research projects have examined the automated analysis of text from captions. Srihari (1991; 1995) did early work on linking information between photographs and their captions, to determine, for example, which person's face in a newspaper photograph corresponded to which name in the caption. Shatkay et al. (2006) combined information from images as well as captions to enhance a text categorization algorithm.

Cohen, Murphy, et al. have explored several different aspects of biological text caption analysis. In one piece of work (Cohen et al., 2003) they devised and tested algorithms for parsing the structure of image captions, which are often quite complex, especially when referring to a figure that has multiple images within it. In another effort, they developed tools to extract information relating to subcellular localization by automatically analyzing fluorescence microscope images of cells (Murphy et al., 2003). They later developed methods to extract facts from the captions referring to these images (Cohen et al., 2003).

Liu et al. (2004) collected a set of figures and classified them according to whether or not they depicted schematic representations of protein interactions. They then allowed users to search for a gene name within the figure caption, returning only those figures that fit within the one class (protein interaction schematics) and contained the gene name.

Yu et al. (2006) created a bioscience image taxonomy (consisting of *Gel-Image*, *Graph*, *Image-of-Thing*, *Mix*, *Model*, and *Table*) and used Support Vector Machines to classify the figures, using properties of both the textual captions and the images.

2.2 Figures in Bioscience Article Search

Some bioscience journal publishers provide a service called "SummaryPlus" that allows for display of figures and captions in the description of a particular article, but the interface does not apply to search results listings.[4]

A medical image retrieval and image annotation task have been part of the ImageCLEF competition since 2005 (Muller et al., 2006).[5] The datasets for this competition are clinical images, and the task is to retrieve images relevant to a query such as "Show blood smears that include polymorphonuclear neu-

[2] http://scholar.google.com

[3] The current version of the interface can be seen at http://biosearch.berkeley.edu

[4] Recently a commercial offering by a company called CSA Illustrata was brought to our attention; it claims to use figures and tables in search in some manner, but detailed information is not freely available.

[5] CLEF stands for Cross-language Evaluation Forum; it originally evaluated multi-lingual information retrieval, but has since broadened its mission.

trophils." Thus, the emphasis is on identifying the content of the images themselves.

Yu and Lee (2006) hypothesized that the information found in the figures of a bioscience article are summarized by sentences from that article's abstract. They succeeded in having 119 scientists mark up the abstract of one of their own articles, indicating which sentence corresponded to each figure in the article. They then developed algorithms to link sentences from the abstract to the figure caption content. They also developed and assessed a user interface called BioEx that makes use of this linking information. The interface shows a set of very small image thumbnails beneath each abstract. When the searcher's mouse hovers over the thumbnail, the corresponding sentence from the abstract is highlighted dynamically.

To evaluate BioEx, Yu and Lee (2006) sent a questionnaire to the 119 biologists who had done the hand-labeling linking abstract sentences to images, asking them to assess three different article display designs. The first design looked like the PubMed abstract view. The second augmented the first view with very small thumbnails of figures extracted from the article. The third was the second view augmented with color highlighting of the abstract's sentences. It is unclear if the biologists were asked to do searches over a collection or were just shown a sample of each view and asked to rate it. 35% of the biologists responded to the survey, and of these, 36 out of 41 (88%) preferred the linked abstract view over the other views. (It should be noted that the effort invested in annotating the abstracts may have affected the scientists' view of the design.)

It is not clear, however, whether biologists would prefer to see the caption text itself rather than the associated information from the abstract. The system described did not allow for searching over text corresponding to the figure caption. The system also did not focus on how to design a full text and caption search system in general.

3 Interface Design and Implementation

The Biotext search engine indexes all Open Access articles available at PubMedCentral. This collection consists of more than 150 journals, 20,000 articles and 80,000 figures. The figures are stored locally, and at different scales, in order to be able to present thumbnails quickly. The Lucene search engine[6] is used to index, retrieve, and rank the text (default statistical ranking). The interface is web-based and is implemented in Python and PHP. Logs and other information are stored and queried using MySQL.

Figure 1a shows the results of searching over the caption text in the Caption Figure view. Figure 1b shows the same search in the Caption Figure with additional Thumbnails (CFT) view. Figure 2a-b shows two examples of the Grid view, in which the query terms are searched for in the captions, and the resulting figures are shown in a grid, along with metadata information.[7] The Grid view may be especially useful for seeing commonalities among topics, such as all the phylogenetic trees that include a given gene, or seeing all images of embryo development of some species.

The next section describes the study participants' reaction to these designs.

4 Pilot Usability Study

The design of search user interfaces is difficult; the evidence suggests that most searchers are reluctant to switch away from something that is familiar. A search interface needs to offer something qualitatively better than what is currently available in order to be acceptable to a large user base (Hearst, 2006).

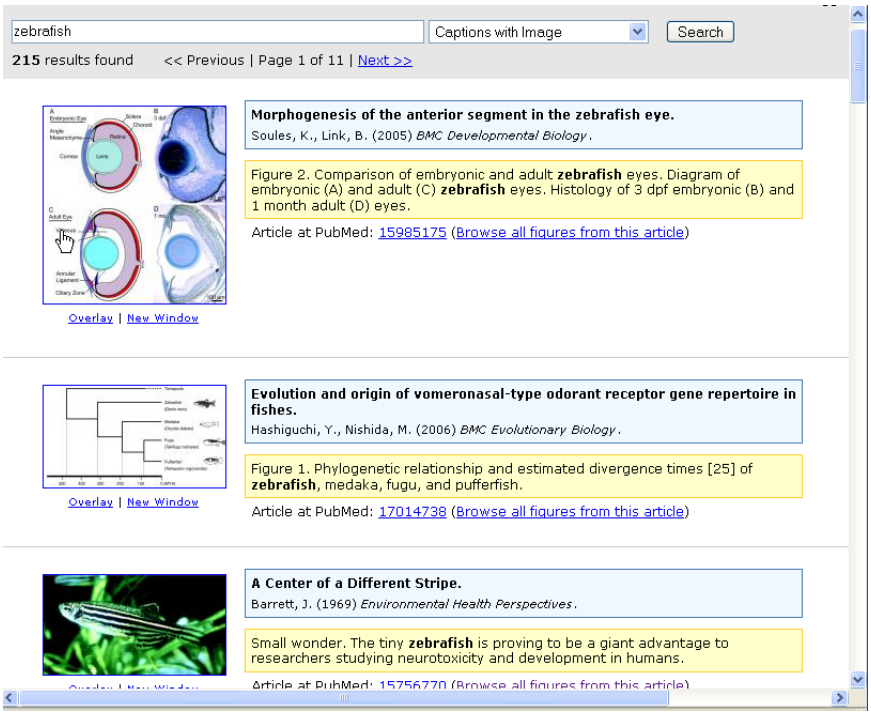
Because text search requires the display of text, results listings can quickly obtain an undesirably cluttered look, and so careful attention to detail is required in the elements of layout and graphic design. Small details that users find objectionable can render an interface objectionable, or too difficult to use. Thus, when introducing a new search interface idea, great care must be taken to get the details right. The practice of user-centered design teaches how to achieve this goal: first prototype, then test the results with potential users, then refine the design based on their responses, and repeat (Hix and Hartson, 1993; Shneiderman and Plaisant, 2004).

Before embarking on a major usability study to determine if a new search interface idea is a good one, it is advantageous to run a series of pilot studies to determine which aspects of the design work,

[6] http://lucene.apache.org

[7] These screenshots represent the system as it was evaluated. The design has subsequently evolved and changed.

(a)



(b)

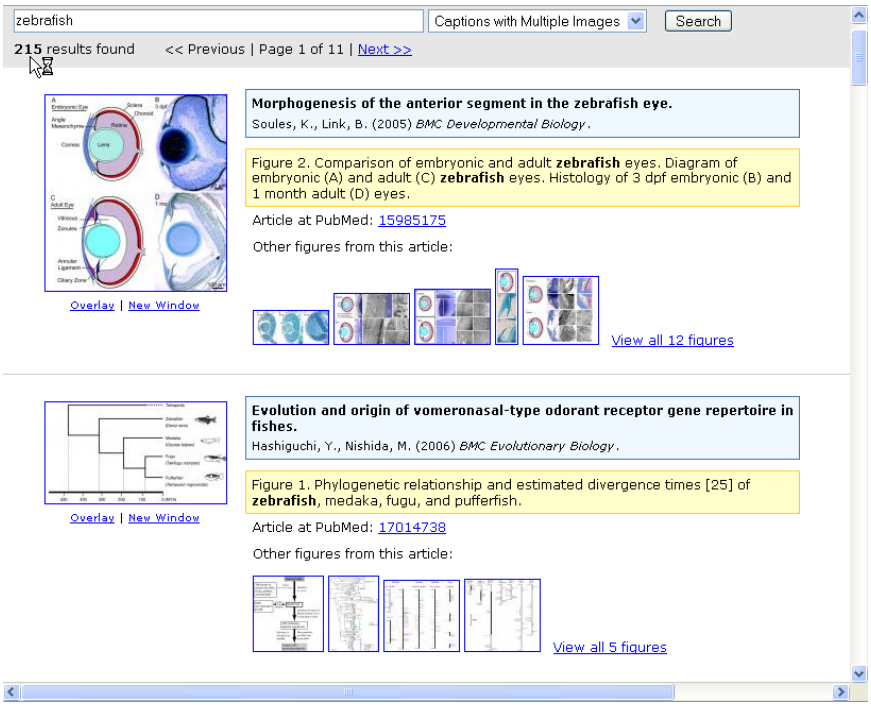
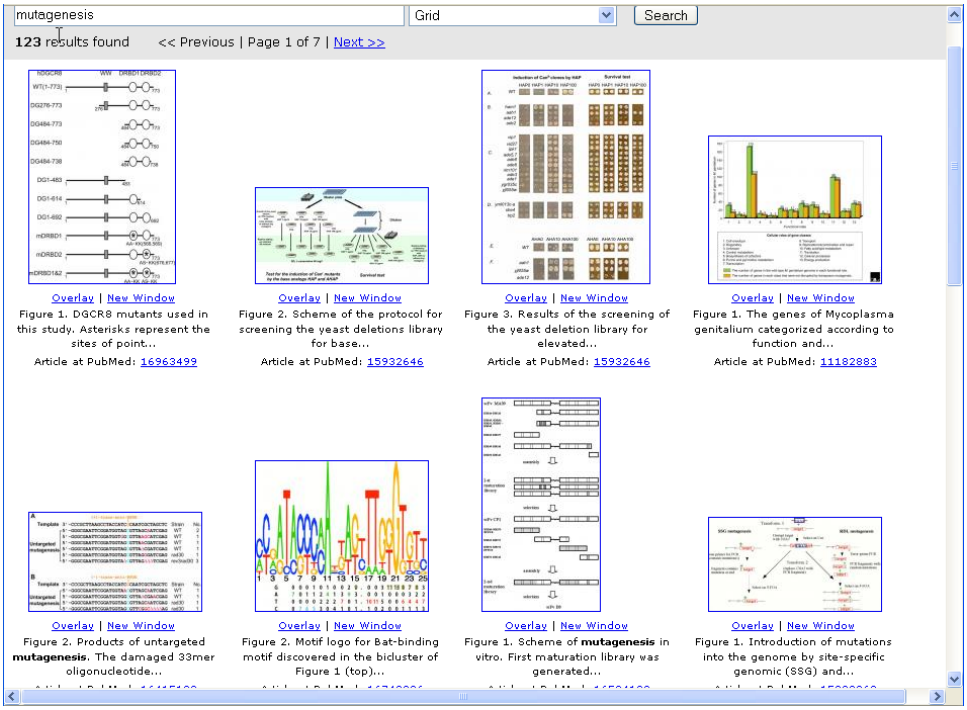


Figure 1: Search results on a query of *zebrafish* over the captions within the articles with (a) CF view, and (b) CFT view. The thumbnail is shown to the left of a blue box containing the bibliographic information above a yellow box containing the caption text. The full-size view of the figure can be overlaid over the current page or in a new browser window. In (b) the first few figures are shown as mini-thumbnails in a row below the caption text with a link to view all the figures and captions.

(a)



(b)

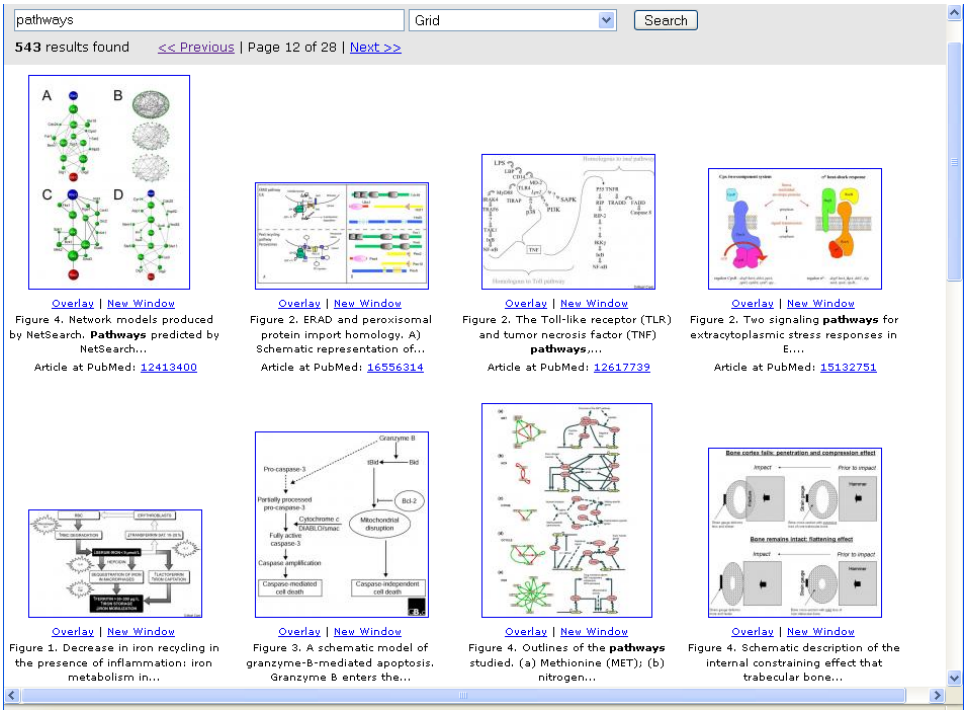


Figure 2: Grid views of the first sets of figures returned as the result of queries for (a) *mutagenesis* and for (b) *pathways* over captions in the Open Access collection.

ID	status	sex	lit search	area(s) of specialization
1	undergrad	F	monthly	organic chemistry
2	graduate	F	weekly	genetics / molecular bio.
3	other	F	rarely	medical diagnostics
4	postdoc	M	weekly	neurobiology, evolution
5	graduate	F	daily	evolutionary bio., entomology
6	undergrad	F	weekly	molecular bio., biochemistry
7	undergrad	F	monthly	cell developmental bio.
8	postdoc	M	daily	molecular / developmental bio.

Table 1: Participant Demographics. Participant 3 is an unemployed former lab worker.

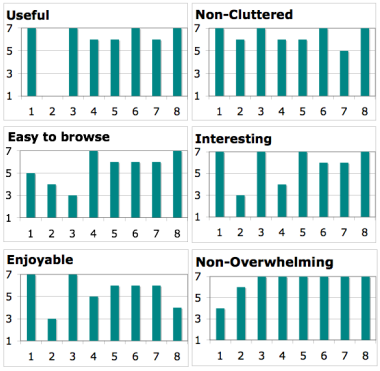


Figure 3: Likert scores on the CF view. X-axis: participant ID, y-axis: Likert scores: 1 = strongly disagree, 7 = strongly agree. (Scale reversed for questionnaire-posed *cluttered* and *overwhelming*.)

which do not, make adjustments, and test some more. Once the design has stabilized and is receiving nearly uniform positive feedback from pilot study participants, then a formal study can be run that compares the novel idea to the state-of-the-art, and evaluates hypotheses about which features work well for which kinds of tasks.

The primary goal of this pilot study was to determine if biological researchers would find the idea of caption search and figure display to be useful or not. The secondary goal was to determine, should caption search and figure display be useful, how best to support these features in the interface. We want to retain those aspects of search interfaces that are both familiar and useful, and to introduce new elements in such a way as to further enhance the search experience without degrading it.

4.1 Method

We recruited participants who work in our campus' main biology buildings to participate in the study. None of the participants were known to us in advance. To help avoid positive bias, we told participants that we were evaluating a search system, but did not mention that our group was the one who was designing the system. The participants all had strong interests in biosciences; their demographics are shown in Table 1.

Each participant's session lasted approximately one hour. First, they were told the purpose of the study, and then filled out an informed consent form and a background questionnaire. Next, they used the search interfaces (the order of presentation was varied). Before the use of each search interface, we explained the idea behind the design. The participant then used the interface to search on their own queries for about 10 minutes, and then filled out a questionnaire describing their reaction to that design. After viewing all of the designs, they filled out a post-study questionnaire where they indicated whether or not they would like to use any of the designs in their work, and compared the design to PubMed-type search.

Along with these standardized questions, we had open discussions with participants about their reactions to each view in terms of design and content. Throughout the study, we asked participants to assume that the new designs would eventually search over the entire contents of PubMed and not just the Open Access collection.

We showed all 8 participants the Caption with Figure (CF) view (see Figure 1a), and Caption with Figure and additional Thumbnails (CFT) (see Figure 1b), as we didn't know if participants would want to see additional figures from the caption's paper.[8] We did not show the first few participants the Grid view, as we did not know how the figure/caption search would be received, and were worried about overwhelming participants with new ideas. (Usability study participants can become frustrated if exposed to too many options that they find distasteful or confusing.) Because the figure search did receive pos-

[8]We also experimented with showing full text search to the first five participants, but as that view was problematic, we discontinued it and substituted a title/abstract search for the remaining three participants. These are not the focus of this study and are not discussed further here.

itive reactions from 3 of the first 4 participants, we decided to show the Grid view to the next 4.

4.2 Results

The idea of caption search and figure display was very positively perceived by all but one participant. 7 out of 8 said they would want to use either CF or CFT in their bioscience journal article searches. Figure 3 shows Likert scores for CF view.

The one participant (number 2) who did not like CF nor CFT thought that the captions/figures would not be useful for their tasks, and preferred seeing the articles' abstracts. Many participants noted that caption search would be better for some tasks than others, where a more standard title & abstract or full-text search would be preferable. Some participants said that different views serve different roles, and they would use more than one view depending on the goal of their search. Several suggested combining abstract and figure captions in the search and/or the display. (Because this could lead to search results that require a lot of scrolling, it would probably be best to use modern Web interface technologies to dynamically expand long abstracts and captions.) When asked for their preference versus PubMed, 5 out of 8 rated at least one of the figure searches above PubMed's interface. (In some cases this may be due to a preference for the layout in our design as opposed to entirely a preference for caption search.)

Two of the participants preferred CFT to CF; the rest thought CFT was too busy. It became clear through the course of this study that it would be best to show all the thumbnails that correspond to a given article as the result of a full-text or abstract-text search interface, and to show only the figure that corresponds to the caption in the caption search view, with a link to view all figures from this article in a new page.

All four participants who saw the Grid view liked it, but noted that the metadata shown was insufficient; if it were changed to include title and other bibliographic data, 2 of the 4 who saw Grid said they would prefer that view over the CF view. Several participants commented that they have used Google Images to search for images but they rarely find what they are looking for. They reacted very positively to the idea of a Google Image-type system specialized to biomedical images. One participant went so far as to open up Google Image search and compare the results directly, finding the caption search to be preferable.

All participants favored the ability to browse all figures from a paper once they find the abstract or one of the figures relevant to their query. Two participants commented that if they were looking for general concepts, abstract search would be more suitable but for a specific method, caption view would be better.

4.3 Suggestions for Redesign

All participants found the design of the new views to be simple and clear. They told us that they generally want information displayed in a simple manner, with as few clicks needed as possible, and with as few distracting links as possible. Only a few additional types of information were suggested from some participants: display, or show links to, related papers and provide a link to the full text PDF directly in the search results, as opposed to having to access the paper via PubMed.

Participants also made clear that they would often want to start from search results based on title and abstract, and then move to figures and captions, and from there to the full article, unless they are doing figure search explicitly. In that case, they want to start with CF or Grid view, depending on how much information they want about the figure at first glance.

They also wished to have the ability to sort the results along different criteria, including year of publication, alphabetically by either journal or author name, and by relevance ranking. This result has been seen in studies of other kinds of search interfaces as well (Reiterer et al., 2005; Dumais et al., 2003). We have also received several requests for table caption search along with figure caption search.

5 Conclusions and Future Work

The results of this pilot study suggest that caption search and figure display is a very promising direction for bioscience journal article search, especially paired with title/abstract search and potentially with other forms of full-text search. A much larger-scale study must be performed to firmly establish this result, but this pilot study provides insight about how

to design a search interface that will be positively received in such a study. Our results also suggest that web search systems like Google Scholar and Google Images could be improved by showing images from the articles along lines of specialization.

The Grid view should be able to show images grouped by category type that is of interest to biologists, such as heat maps and phylogenetic trees. One participant searched on *pancreas* and was surprised when the top-ranked figure was an image of a machine. This idea underscores the need for BioNLP research in the study of automated caption classification. NLP is needed both to classify images and perhaps also to automatically determine which images are most "interesting" for a given article.

To this end, we are in the process of building a classifier for the figure captions, in order to allow for grouping by type. We have developed an image annotation interface and are soliciting help with hand-labeling from the research community, to build a training set for an automated caption classifier.

In future, we plan to integrate table caption search, to index the text that refers to the caption, along with the caption, and to provide interface features that allow searchers to organize and filter search results according to metadata such as year published, and topical information such as genes/proteins mentioned. We also plan to conduct formal interface evaluation studies, including comparing to PubMed-style presentations.

Acknowledgements: This work was supported in part by NSF DBI-0317510. We thank the study participants for their invaluable help.

References

H. Chen and B.M. Sharp. 2004. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics*, 5(147).

W.W. Cohen, R. Wang, and R.F. Murphy. 2003. Understanding captions in biomedical publications. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 499–504.

S. Dumais, E. Cutrell, J.J. Cadiz, G. Jancke, R. Sarin, and D.C. Robbins. 2003. Stuff I've seen: a system for personal information retrieval and re-use. *Proceedings of SIGIR 2003*, pages 72–79.

M. Hearst. 2006. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR Workshop on Faceted Search*, Seattle, WA.

W. Hersh, A. Cohen, P. Roberts, and Rekapalli H. K. 2006. TREC 2006 genomics track overview. *The Fifteenth Text Retrieval Conference*.

L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. 2005. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6:1.

D. Hix and H.R. Hartson. 1993. *Developing user interfaces: ensuring usability through product & process*. John Wiley & Sons, Inc. New York, NY, USA.

R. Hoffmann and A. Valencia. 2004. A gene network for navigating the literature. *Nature Genetics*, 36(664).

F. Liu, T-K. Jenssen, V. Nygaard, J. Sack, and E. Hovig. 2004. FigSearch: a figure legend indexing and classification system. *Bioinformatics*, 20(16):2880–2882.

H. Muller, T. Deselaers, T. Lehmann, P. Clough, E. Kim, and W. Hersh. 2006. Overview of the ImageCLEF 2006 Medical Image Retrieval Tasks. In *Working Notes for the CLEF 2006 Workshop*.

R.F. Murphy, M. Velliste, and G. Porreca. 2003. Robust Numerical Features for Description and Classification of Subcellular Location Patterns in Fluorescence Microscope Images. *The Journal of VLSI Signal Processing*, 35(3):311–321.

B. Rafkind, M. Lee, S.F. Chang, and H. Yu. 2006. Exploring text and image features to classify images in bioscience literature. *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL*, 6:73–80.

H. Reiterer, G. Tullius, and T. M. Mann. 2005. Insyder: a content-based visual-information-seeking system for the web. *International Journal on Digital Libraries*, 5(1):25–41, Mar.

P.K. Shah, C. Perez-Iratxeta, P. Bork, and M.A. Andrade. 2003. Information extraction from full text scientific articles: where are the keywords? *BMC Bioinformatics*, 4(20).

H. Shatkay, N. Chen, and D. Blostein. 2006. Integrating image data into biomedical text categorization. *Bioinformatics*, 22(14):e446.

B. Shneiderman and C. Plaisant. 2004. *Designing the user interface: strategies for effective human-computer interaction, 4/E*. Addison Wesley.

R.K. Srihari. 1991. PICTION: A System that Uses Captions to Label Human Faces in Newspaper Photographs. *Proceedings AAAI-91*, pages 80–85.

RK Srihari. 1995. Automatic indexing and content-based retrieval of captioned images. *Computer*, 28(9):49–56.

A.S. Yeh, L. Hirschman, and A.A. Morgan. 2003. Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup. *Bioinformatics*, 19(1):i331–i339.

H. Yu and M. Lee. 2006. Accessing bioscience images from abstract sentences. *Bioinformatics*, 22(14):e547.

ConText: An Algorithm for Identifying Contextual Features from Clinical Text

Wendy W. Chapman, David Chu, John N. Dowling

Department of Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA
chapman@cbmi.pitt.edu

Abstract

Applications using automatically indexed clinical conditions must account for contextual features such as whether a condition is negated, historical or hypothetical, or experienced by someone other than the patient. We developed and evaluated an algorithm called ConText, an extension of the NegEx negation algorithm, which relies on trigger terms, pseudo-trigger terms, and termination terms for identifying the values of three contextual features. In spite of its simplicity, ConText performed well at identifying negation and hypothetical status. ConText performed moderately at identifying whether a condition was experienced by someone other than the patient and whether the condition occurred historically.

1 Introduction

Natural language processing (NLP) techniques can extract variables from free-text clinical records important for medical informatics applications performing decision support, quality assurance, and biosurveillance [1-6]. Many applications have focused on identifying individual clinical conditions in textual records, which is the first step in making the conditions available to computerized applications. However, identifying individual instances of clinical conditions is not sufficient for many medical informatics tasks—the context surrounding the condition is crucial for integrating the information within the text to determine the clinical state of a patient.

For instance, it is important to understand whether a condition is affirmed or negated, acute or chronic, or mentioned hypothetically. We refer to these as contextual features, because the information is not usually contained in the lexical representation of the clinical condition itself but in the context surrounding the clinical condition. We developed an algorithm called ConText for identifying three contextual features relevant for biosurveillance from emergency department (ED) reports and evaluated its performance compared to physician annotation of the features.

2 Background

2.1 Encoding Contextual Information from Clinical Texts

NLP systems designed to encode detailed information from clinical reports, such as MedLEE [1], MPLUS [7], and MedSyndikate [4], encode contextual features such as negation, uncertainty, change over time, and severity. Over the last ten years, several negation algorithms have been described in the literature [8-12]. Recently, researchers at Columbia University have categorized temporal expressions in clinical narrative text and evaluated a temporal constraint structure designed to model the temporal information for discharge summaries [13, 14].

ConText differs from most other work in this area by providing a stand-alone algorithm that can be integrated with any application that indexes clinical conditions from text.

2.2 Biosurveillance from ED Data

Biosurveillance and situational awareness are imperative research issues in today's world. State-of-the-art surveillance systems rely on chief complaints and ICD-9 codes, which provide limited clinical information and have been shown to perform with only fair to moderate sensitivity [15-18]. ED reports are a timely source of clinical informa-

BioNLP 2007: Biological, translational, and clinical language processing, pages 81–88,
Prague, June 2007.

tion that may be useful for syndromic surveillance. We are developing NLP-based methods for identifying clinical conditions from ED reports.

2.3 SySTR

We are developing an NLP application called SySTR (Syndromic Surveillance from Textual Records). It currently uses free-text descriptions of clinical conditions in ED reports to determine whether the patient has an acute lower respiratory syndrome. We previously identified 55 clinical conditions (e.g. cough, pneumonia, oxygen desaturation, wheezing) relevant for determining whether a patient has an acute lower respiratory condition [19]. SySTR identifies instances of these 55 clinical conditions in ED reports to determine if a patient has an acute lower respiratory syndrome. SySTR has four modules:

(1) Index each instance of the 55 clinical conditions in an ED report;
(2) For each indexed instance of a clinical condition, assign values to three contextual features;
(3) Integrate the information from indexed instances to determine whether each of the 55 conditions are *acute*, *chronic*, or *absent*;
(4) Use the values of the 55 conditions to determine whether a patient has an acute lower respiratory syndrome.

We built SySTR on top of an application called caTIES [20], which comprises a GATE pipeline of processing resources (http://gate.ac.uk/). Module 1 uses MetaMap [5] to index UMLS concepts in the text and then maps the UMLS concepts to the 55 clinical conditions. For instance, Module 1 would identify the clinical condition Dyspnea in the sentence "Patient presents with a 3 day history of shortness of breath." For each instance of the 55 conditions identified by Module 1, Module 2 assigns values to three contextual features: Negation (*negated, affirmed*); Temporality (*historical, recent, hypothetical*); and Experiencer (*patient, other*). For the sentence above, Module 2 would assign Dyspnea the following contextual features and their values: Negation—*affirmed*; Temporality—*recent*; Experiencer—*patient*. Module 3, as described in Chu and colleagues [21], resolves contradictions among multiple instances of clinical conditions, removes conditions not experienced by the patient, and assigns a final value of *acute*, *chronic*, or *absent* to each of the 55 conditions. Module 4 uses machine learning models to determine whether a patient has acute lower respiratory syndrome based on values of the conditions.

The objective of this study was to evaluate an algorithm for identifying the contextual information generated by Module 2.

3 Methods

We developed an algorithm called ConText for determining the values for three contextual features of a clinical condition: Negation, Temporality, and Experiencer. The same algorithm is applied to all three contextual features and is largely based on a regular expression algorithm for determining whether a condition is negated or not (NegEx [9]). ConText relies on trigger terms, pseudo-trigger terms, and scope termination terms that are specific to the type of contextual feature being identified. Below we describe the three contextual features addressed by the algorithm, details of how ConText works, and our evaluation of ConText.

3.1 Three Contextual Features

Determining whether a patient had an acute episode of a clinical condition, such as cough, potentially involves information described in the context of the clinical condition in the text. We performed a pilot study to learn which contextual features affected classification of 55 clinical conditions as *acute*, *chronic*, or *absent* [21]. The pilot study identified which contextual features were critical for our task and reduced the number of values we initially used.

The contextual features for each indexed clinical condition are assigned default values. ConText changes the values if the condition falls within the scope of a relevant trigger term. Below, we describe the contextual features (default values are in parentheses).

(1) **Negation** (*affirmed*): ConText determines whether a condition is negated, as in "No fever."
(2) **Temporality** (*recent*): ConText can change Temporality to *historical* or *hypothetical*. In its current implementation, *historical* is defined as beginning at least 14 days before the visit to the ED, but the algorithm can easily be modified to change the length of time. ConText would mark Fever in "Patient should return if she develops fever" as *hypothetical*.

(3) **Experiencer** (*patient*): ConText assigns conditions ascribed to someone other than the patient an Experiencer of *other*, as in "The patient's father has a history of CHF."

3.2 Contextual Feature Algorithm

As we examined how the contextual features were manifested in ED reports, we discovered similar patterns for all features and hypothesized that an existing negation algorithm, NegEx [9], may be applicable for all three features.

NegEx uses two regular expressions (RE) to determine whether an indexed condition is negated:

RE1: <trigger term> <5w> <indexed term>
RE2: <indexed term> <5w> <trigger term>

<5w> represents five words (a word can be a single word or a UMLS concept), and the text matched by this pattern is called the scope. NegEx relies on three types of terms to determine whether a condition is negated: trigger terms, pseudo-trigger terms, and termination terms. Trigger terms such as "no" and "denies" indicate that the clinical conditions that fall within the scope of the trigger term should be negated. Pseudo-trigger terms, such as "no increase," contain a negation trigger term but do not indicate negation of a clinical concept. A termination term such as "but" can terminate the scope of the negation before the end of the window, as in "She denies headache but complains of dizziness."

ConText is an expansion of NegEx. It relies on the same basic algorithm but applies different term lists and different windows of scope depending on the contextual feature being annotated.

3.3 ConText Term Lists

Each contextual feature has a unique set of trigger terms and pseudo-trigger terms, as shown in Table 1. The complete list of terms can be found at http://web.cbmi.pitt.edu/chapman/ConText.html. Most of the triggers apply to RE1, but a few (marked in table) apply to RE2. ConText assigns a default value to each feature, then changes that value if a clinical condition falls within the scope of a relevant trigger term.

Although trigger terms are unique to the contextual feature being identified, termination terms

Table 1. Examples of trigger and pseudo-trigger terms for the three contextual features. If all terms are not represented in the table, we indicate the number of terms used by ConText in parentheses.

Temporality (default = *recent*)

Trigger terms for *hypothetical*	Pseudo-trigger terms	Trigger terms for *historical*	Pseudo-trigger terms (10)
if	if negative	General triggers	history, physical
return		history	history taking
should [he|she]		previous^	poor history
should there		History Section title^^	history and examination
should the patient		Temporal Measurement triggers^^^	history of present illness
as needed		<time> of	social history
come back [for|to]		[for|over] the [last|past] <time>	family history
		since (last) [day-of-week|week|month|season|year]	sudden onset of

Experiencer (default = *patient*)

Trigger terms for *other* (12)	Pseudo-trigger terms
father('s)	
mother('s)	
aunt('s)	

Negation (default = *affirmed*)

Trigger terms for *negated* (125)	Pseudo-trigger terms (16)
no	no increase
not	not extend
denies	gram negative
without	

^ the scope for "previous" only extends one term forward (e.g., "for previous headache")
^^Currently the only history section title we use is PAST MEDICAL HISTORY.
^^^ <time> includes the following regular expression indicating a temporal quantification: x[-|space] [day(s)|hour(s)|week(s)|month(s)|year(s)]. x = any digit; words in brackets are disjunctions; items in parentheses are optional. The first two temporal measurement triggers are used with RE1; the third is used with RE2. For our current application, a condition lasting 14 days or more is considered *historical*.

may be common to multiple contextual features. For instance, a termination term indicating that the physician is speaking about the patient can indicate termination of scope for the features Temporality and Experiencer. In the sentence "History of COPD, presenting with shortness of breath," the trigger term "history" indicates that COPD is *historical*, but the term "presenting" terminates the scope of the temporality trigger term, because the physician is now describing the current patient visit. Therefore, the condition Dyspnea ("shortness of breath") should be classified as *recent*. Similarly, in the sentence "Mother has CHF and patient presents with chest pain," Experiencer for CHF should be *other*, but Experiencer for Chest Pain should be *patient*.

We compiled termination terms into conceptual groups, as shown in Table 2.

Table 2. ConText's termination terms. Column 1 lists the type of termination term, the number of terms used by Context, and the contextual feature values using that type of termination term. Column 2 gives examples of the terms.

Type of Term	Examples
Patient (5) Temporal (hypothetical) Experiencer (other)	Patient, who, his, her, patient's
Presentation (12) Temporal (historical) Experiencer (other)	Presents, presenting, complains, was found, states, reports, currently, today
Because (2) Temporal (hypothetical)	Since, because
Which (1) Experiencer (other)	Which
ED (2) Temporal (historical)	Emergency department, ED
But (8) Negation (negated)	But, however, yet, though, although, aside from

3.4 ConText Algorithm

The input to ConText is an ED report with instances of the 55 clinical concepts already indexed. For each clinical condition, ConText assigns values to the three contextual features. ConText's algorithm is as follows[1]:

Go to first trigger term in sentence
  If term is a pseudo-trigger term,
    Skip to next trigger term
Determine scope of trigger term
  If termination term within scope,
    Terminate scope before termination term
Assign appropriate contextual feature value to all indexed clinical concepts within scope.

The scope of a trigger term depends on the contextual feature being classified. The default scope includes all text following the indexed condition until the end of the sentence. Thus, in the sentence "He should return for fever" the scope of the Temporality (hypothetical) trigger term "return" includes the segment "for fever," which includes an indexed condition Fever. The default scope is overridden in a few circumstances. First, as described above, the scope can be terminated by a relevant termination term. Second, if the trigger term is a <section title>, the scope extends throughout the entire section, which is defined previous to ConText's processing. Third, a trigger term itself can require a different scope. The Temporality (historical) term "previous" only extends one term forward in the sentence.

3.5 Evaluation

We evaluated ConText's ability to assign correct values to the three contextual features by comparing ConText's annotations with annotations made by a physician.

Setting and Subjects. The study was conducted on reports for patients presenting to the University of Pittsburgh Medical Center Presbyterian Hospital ED during 2002. The study was approved by the University of Pittsburgh's Institutional Review Board. We randomly selected 120 reports for patients with respiratory-related ICD-9 discharge diagnoses for manual annotation. For this study, we used 30 reports as a development set and 90 reports as a test set. In addition to the annotated development set, we used a separate set of 100 unannotated ED reports to informally validate our term lists.

Reference Standard. A physician board-certified in internal medicine and infectious diseases with 30 years of experience generated manual annotations for the development and test reports. He used GATE (http://gate.ac.uk/) to highlight every indi-

[1] This algorithm applies to RE1. The algorithm for RE2 is the same, except that it works backwards from the trigger term and does not look for pseudo-trigger terms.

vidual annotation in the text referring to any of the 55 clinical conditions. For every annotation, he assigned values to the three contextual features, as shown in Figure 1.

Previous experience in annotating the 55 conditions showed that a single physician was inadequate for generating a reliable reference standard [19]. The main mistake made by a single physician was not marking a concept that existed in the text. We used NLP-assisted review to improve physician annotations by comparing the single physician's annotations to those made by SySTR. The physician reviewed disagreements and made changes to his original annotations if he felt his original annotation was incorrect. A study by Meystre and Haug [22] used a similar NLP-assisted review methodology and showed that compared to a reference standard not using NLP-assisted review, their system had higher recall and the same precision.

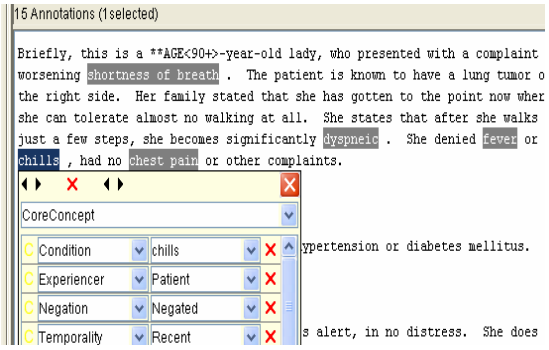


Figure 1. When the physician highlights text, GATE provides a drop-down menu to select the Clinical Condition and the values of the Contextual Features.

Outcome Measures. For each contextual feature assigned to an annotation, we compared ConText's value to the value assigned by the reference standard. We classified the feature as a true positive (TP) if ConText correctly changed the condition's default value and a true negative (TN) if ConText correctly left the default value. We then calculated recall and precision using the following formulas:

$$\text{Recall}: \frac{number\ of\ TP}{(number\ of\ TP + number\ of\ FN)}$$

$$\text{Precision}: \frac{number\ of\ TP}{(number\ of\ TP + number\ of\ FP)}$$

For the Temporality feature, we calculated recall and precision separately for the values *historical* and *hypothetical*. We calculated the 95% confidence intervals (CI) for all outcome measures.

4 Results

Using NLP-assisted review, the reference standard physician made several changes to his initial annotations. He indexed an additional 82 clinical conditions and changed the title of the clinical condition for 48 conditions, resulting in a total of 1,620 indexed clinical conditions in the 90 test reports. The reference standard physician also made 35 changes to Temporality values and 4 changes to Negation. The majority of Temporality changes were from *historical* to *recent* (17) and from *hypothetical* to *recent* (12).

Table 3 shows ConText's recall and precision values compared to the reference standard annotations. About half of the conditions were *negated* (773/1620). Fewer conditions were *historical* (95/1620), *hypothetical* (40/1620), or experienced by someone *other* than the patient (8/1620). In spite of low frequency for these contextual feature values, identifying them is critical to understanding a patient's current state. ConText performed best on Negation, with recall and precision above 97%. ConText performed well at assigning the Temporality value *hypothetical*, but less well on the Temporality value *historical*. Experiencer had a small sample size, making results difficult to interpret.

Table 3. Outcome measures for ConText on test set of 90 ED reports.

Feature	TP	TN	FP	FN	Recall 95% CI	Precision 95% CI
Negation	750	824	23	23	97.0 96-98	97.0 96-98
Temporality (*historical*)	66	1499	23	32	67.4 58-76	74.2 64-82
Temporality (*hypothetical*)	33	1578	2	7	82.5 68-91	94.3 81-98
Experiencer	4	1612	0	4	50.00 22-78	100 51-100

5 Discussion

We evaluated an extension of the NegEx algorithm for determining the values of two additional contextual features—Temporality and Experiencer. ConText performed with very high recall and precision when determining whether a condition was negated, and demonstrated moderate to high performance on the other features.

We performed an informal error analysis, which not only isolates ConText's errors but also points out future research directions in contextual feature identification.

5.1 Negation

ConText's negation identification performed substantially better than NegEx's published results [9], even though ConText is very similar to NegEx and uses the same trigger terms. Several possible explanations exist for this boost in performance. First, our study evaluated negation identification in ED reports, whereas the referenced study on NegEx applied to discharge summaries. Second, ConText only applied to 55 clinical conditions, rather than the large set of UMLS concepts in the NegEx study. Third, the conditions indexed by SySTR that act as input to ConText are sometimes negated or affirmed before ConText sees them. For some conditions, SySTR addresses internal negation in a word (e.g., "afebrile" is classified as Fever with the Negation value *negated*). Also, SySTR assigns Negation values to some conditions with numeric values, such as negating Tachycardia from "pulse rate 75." Fourth, ConText does not use NegEx's original scope of five words, but extends the scope to the end of the sentence. It would be useful to compare ConText's scope difference directly against NegEx to determine which scope assignment works better, but our results suggest the increased scope may work well for ED reports.

ConText's errors in assigning the Negation value were equally distributed between FN's and FP's (23 errors each). Some false negatives resulted from missing trigger terms (e.g., "denying"). Several false negatives resulted from the interaction between ConText and SySTR's mapping rules. For example, in the sentence "chest wall is without tenderness," SySTR maps the UMLS concepts for "chest wall" and "tenderness" to the condition Chest Wall Tenderness. In such a case, the negation trigger term "without" is caught between the two UMLS concepts. Therefore, RE1 does not match, and ConText does not change the default from *affirmed*. False positive negations resulted from our not integrating the rule described in NegEx that a concept preceded by a definite article should not be negated [23] (e.g., "has not been on steroids for his asthma") and from descriptions in the text whose Negation status is even difficult for humans to determine, such as "no vomiting without having the cough" and "patient does not know if she has a fever."

5.2 Temporality

Historical. ConText identified *historical* conditions with 67% sensitivity and 74% precision. Identifying historical conditions appears simple on the surface, but is a complex problem. The single trigger term "history" is used for many of the historical conditions, but the word "history" is a relative term that can indicate a history of years (as in "history of COPD") or of only a few days (as in "ENT: No history of nasal congestion"). The error analysis showed that ConText is missing trigger terms that act equivalently to the word "history" such as "in the past" ("has not been on steroids in the past for his asthma") and "pre-existing" ("pre-existing shortness of breath").

Some conditions that the reference standard classified as *historical* had no explicit trigger in the text, as in the sentence "When he sits up in bed, he develops pain in the chest." It may be useful to implement rules involving verb tense for these cases.

The most difficult cases for ConText were those with temporal measurement triggers. The few temporal quantifier patterns we used were fairly successful, but the test set contained multiple variations on those quantifiers, and a new dataset would probably introduce even more variations. For instance, ConText falsely classified Non-pleuritic Chest Pain as *historical* in "awoken at approximately 2:45 with chest pressure," because ConText's temporal quantifiers do not account for time of the day. Also, even though ConText's temporal quantifiers include the pattern "last x weeks," x represents a digit and thus didn't match the phrase "intermittent cough the last couple of weeks."

We were hoping that identifying historical conditions would not require detailed modeling of temporal information, but our results suggest otherwise. We will explore the temporal categories derived by Hripcsak and Zhou [13] for discharge summaries to expand ConText's ability to identify temporal measurement triggers.

Hypothetical. ConText demonstrated 83% recall and 94% precision when classifying a condition as *hypothetical* rather than *recent*. Again, missing trigger terms (e.g., "returning" and "look out for") and termination terms (e.g., "diagnosis") caused errors. The chief cause of false negatives was ter-

minating the scope of a trigger term too early. For instance, in the sentence "She knows to return to the ED if she has anginal type chest discomfort which was discussed with her, shortness of breath, and peripheral edema" the scope of the trigger "return" was terminated by "her." The major limitation of regular expressions is evident in this example in which "her" is part of a relative clause modifying "chest discomfort," not "shortness of breath."

5.3 Experiencer

ConText's ability to identify an experiencer other than the patient suffered from low prevalence. In the test set of 90 reports, only 8 of the 1620 conditions were experienced by someone other than the patient, and ConText missed half of them. Two of the false negatives came from not including the trigger term "family history." A more difficult error to address is recognizing that bronchitis is experienced by someone other than the patient in "…due to the type of bronchitis that is currently being seen in the community." ConText made no false positive classifications for Experiencer.

5.4 Limitations and Future Work

Some of ConText's errors can be resolved by refining the trigger and termination terms. However, many of the erroneous classifications are due to complex syntax and semantics that cannot be handled by simple regular expressions. Determining the scope of trigger terms in sentences with relative clauses and coordinated conjunctions is especially difficult. We believe ConText's approach involving trigger terms, scope, and termination terms is still a reasonable model for this problem and hope to improve ConText's ability to identify scope with syntactic information.

A main limitation of our evaluation was the reference standard, which was comprised of a single physician. We used NLP-assisted review to increase the identification of clinical conditions and decrease noise in his classifications. It is possible that the NLP-assisted review biased the reference standard toward ConText's classifications, but the majority of changes made after NLP-assisted review involved indexing the clinical conditions, rather than changing the values of the contextual features. Moreover, most of the changes to contextual feature values involved a change in our annotation schema after the physician had completed his first round of annotations. Specifically, we allowed the physician to use the entire report to determine whether a condition was *historical*, which caused him to mark recent exacerbations of historical conditions as *historical*. A second physician is in the process of annotating the test set. The two physicians will come to consensus on their classifications in generating a new reference standard.

How good contextual feature identification has to be depends largely on the intended application. We tested SySTR's ability to determine whether the 55 clinical conditions were *acute*, *chronic*, or *absent* on a subset of 30 test reports [24]. SySTR made 51 classification errors, 22 of which were due to ConText's mistakes. In spite of the errors, SySTR demonstrated a kappa of 0.85 when compared to physician classifications, suggesting that because of redundancy in clinical reports, ConText's mistakes may not have a substantial adverse effect on SySTR's final output.

5.5 Conclusion

We evaluated a regular-expression-based algorithm for determining the status of three contextual features in ED reports and found that ConText performed very well at identifying negated conditions, fairly well at determining whether conditions were hypothetical or historical, and moderately well at determining whether a condition was experienced by someone other than the patient. ConText's algorithm is based on the negation algorithm NegEx, which is a frequently applied negation algorithm in biomedical informatics applications due to its simplicity, availability, and generalizability to various NLP applications. Simple algorithms for identifying contextual features of indexed conditions is important in medical language processing for improving the accuracy of information retrieval and extraction applications and for providing a baseline comparison for more sophisticated algorithms. ConText accepts any indexed clinical conditions as input and thus may be applicable to other NLP applications. We do not know how well ConText will perform on other report types, but see similar contextual features in discharge summaries, progress notes, and history and physical exams. Currently, ConText only identifies three contextual features, but we hope to extend the algorithm to other features in the future, such as whether a condition is mentioned as a radiology finding or as a diagnosis (e.g., Pneumonia).

Over and above negation identification, which can be addressed by NegEx or other algorithms, ConText could be useful for a variety of NLP tasks, including flagging historical findings and eliminating indexed conditions that are hypothetical or were not experienced by the patient. Ability to modify indexed conditions based on their contextual features can potentially improve precision in biosurveillance, real-time decision support, and information retrieval.

Acknowledgments. This work was supported by NLM grant K22 LM008301, "Natural language processing for respiratory surveillance."

References

1. Friedman C. A broad-coverage natural language processing system. Proc AMIA Symp 2000:270-4.
2. Fiszman M, Chapman WW, Aronsky D, Evans RS, Haug PJ. Automatic detection of acute bacterial pneumonia from chest X-ray reports. J Am Med Inform Assoc 2000;7(6):593-604.
3. Taira R, Bashyam V, Kangarloo H. A field theory approach to medical natural language processing. IEEE Transactions in Inform Techn in Biomedicine 2007;11(2).
4. Hahn U, Romacker M, Schulz S. MEDSYNDIKATE-a natural language system for the extraction of medical information from findings reports. Int J Med Inf 2002;67(1-3):63-74.
5. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp 2001:17-21.
6. Hazlehurst B, Frost HR, Sittig DF, Stevens VJ. MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record. J Am Med Inform Assoc 2005;12(5):517-29.
7. Christensen L, Haug PJ, Fiszman M. MPLUS: a probabilistic medical language understanding system. Proc Workshop on Natural Language Processing in the Biomedical Domain 2002:29-36.
8. Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J Am Med Inform Assoc 2001;8(6):598-609.
9. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 2001;34(5):301-10.
10. Elkin PL, Brown SH, Bauer BA, Husser CS, Carruth W, Bergstrom LR, et al. A controlled trial of automated classification of negation from clinical notes. BMC Med Inform Decis Mak 2005;5(1):13.
11. Herman T, Matters M, Walop W, Law B, Tong W, Liu F, et al. Concept negation in free text components of vaccine safety reports. AMIA Annu Symp Proc 2006:1122.
12. Huang Y, Lowe HJ. A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. J Am Med Inform Assoc 2007.
13. Hripcsak G, Zhou L, Parsons S, Das AK, Johnson SB. Modeling electronic discharge summaries as a simple temporal constraint satisfaction problem. J Am Med Inform Assoc 2005;12(1):55-63.
14. Zhou L, Melton GB, Parsons S, Hripcsak G. A temporal constraint structure for extracting temporal information from clinical narrative. J Biomed Inform 2005.
15. Chapman WW, Dowling JN, Wagner MM. Classification of emergency department chief complaints into seven syndromes: a retrospective analysis of 527,228 patients. Ann Emerg Med 2005;46(5):445-455.
16. Ivanov O, Wagner MM, Chapman WW, Olszewski RT. Accuracy of three classifiers of acute gastrointestinal syndrome for syndromic surveillance. Proc AMIA Symp 2002:345-9.
17. Chang HG, Cochrane DG, Tserenpuntsag B, Allegra JR, Smith PF. ICD9 as a surrogate for chart review in the validation of a chief complaint syndromic surveillance system. In: Syndromic Surveillance Conference Seattle, Washington; 2005.
18. Beitel AJ, Olson KL, Reis BY, Mandl KD. Use of emergency department chief complaint and diagnostic codes for identifying respiratory illness in a pediatric population. Pediatr Emerg Care 2004;20(6):355-60.
19. Chapman WW, Fiszman M, Dowling JN, Chapman BE, Rindflesch TC. Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap. Medinfo 2004;2004:487-91.
20. Mitchell KJ, Becich MJ, Berman JJ, Chapman WW, Gilbertson J, Gupta D, et al. Implementation and evaluation of a negation tagger in a pipeline-based system for information extraction from pathology reports. Medinfo 2004;2004:663-7.
21. Chu D, Dowling JN, Chapman WW. Evaluating the effectiveness of four contextual features in classifying annotated clinical conditions in emergency department reports. AMIA Annu Symp Proc 2006:141-5.
22. Meystre S, Haug PJ. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. J Biomed Inform 2006;39(6):589-99.
23. Goldin I, Chapman WW. Learning to detect negation with 'not' in medical texts. In: Proc Workshop on Text Analysis and Search for Bioinformatics at the 26th Annual International ACM SIGIR Conference (SIGIR-2003); 2003.
24. Chu D. Clinical feature extraction from emergency department reports for biosurveillance [Master's Thesis]. Pittsburgh: University of Pittsburgh; 2007.

BioNoculars: Extracting Protein-Protein Interactions from Biomedical Text

Amgad Madkour, *Kareem Darwish, Hany Hassan, Ahmed Hassan, Ossama Emam
Human Language Technologies Group
IBM Cairo Technology Development Center
P.O.Box 166 El-Ahram, Giza, Egypt
{amadkour,hanyh,hasanah,emam}@eg.ibm.com,*kareem@darwish.org

Abstract

The vast number of published medical documents is considered a vital source for relationship discovery. This paper presents a statistical unsupervised system, called BioNoculars, for extracting protein-protein interactions from biomedical text. BioNoculars uses graph-based mutual reinforcement to make use of redundancy in data to construct extraction patterns in a domain independent fashion. The system was tested using MEDLINE abstract for which the protein-protein interactions that they contain are listed in the database of interacting proteins and protein-protein interactions (DIPPPI). The system reports an F-Measure of 0.55 on test MEDLINE abstracts.

1 Introduction

With the ever-increasing number of published biomedical research articles and the dependency of new research and previously published research, medical researchers and practitioners are faced with the daunting prospect of reading through hundreds or possibly thousands of research articles to survey advances in areas of interest. Much work has been done to ease access and discovery of articles that match the interest of researchers via the use of search engines such as PubMed, which provides search capabilities over MEDLINE, a collection of more than 15 million journal paper abstracts maintained by the National Library of Medicine (NLM). However, with the addition of abstracts from more than 5,000 medical journals to MEDLINE every year, the number of articles containing information that is pertinent to users needs has grown considerably. These 5,000 journals constitute only a subset of the published biomedical research. Further, medical articles often contain redundant information and only subsections of articles are typically of direct interest to researchers. More advanced information extraction tools have been developed to effectively distill medical articles to produce key pieces of information from articles while attempting to eliminate redundancy. These tools have focused on areas such as protein-protein interaction, gene-disease relationship, and chemical-protein interaction (Chun et al., 2006). Many of these tools have been used to extract key pieces of information from MEDLINE. Most of the reported information extraction approaches use sets of handcrafted rules in conjunction with manually curated dictionaries and ontologies.

This paper presents a fully unsupervised statistical technique to discover protein-protein interaction based on automatically discoverable repeating patterns in text that describe relationships. The paper is organized as follows: section 2 surveys related work; section 3 describes BioNoculars; Section 4 describes the employed experimental setup; section 5 reports and comments on experimental results; and section 6 concludes the paper.

2 Background

The background will focus primarily on the tagging of Biomedical Named Entities (BNE), such genes, gene-products, proteins, and chemicals and the Ex-

BioNLP 2007: Biological, translational, and clinical language processing, pages 89–96,
Prague, June 2007.

traction of protein-protein interactions from text.

2.1 BNE Tagging

Concerning BNE tagging, the most common approaches are based on hand-crafted rules, statistical classifiers, or a hybrid of both (usually in conjunction with dictionaries of BNE). Rule-based systems (Fukuda et al., 1998; Hanisch et al., 2003; Yamamoto et al., 2003) that use dictionaries tend to exhibit high precision in tagging named entities but generally with lower tagging recall. They tend to lag the latest published research and are sensitive to the expression of the named entities. Dictionaries of BNE are typically laborious and expensive to build, and they are dependant on nomenclatures and specific species. Statistical approaches (Collier et al., 2000; Kazama et al., 2002; Settles, 2004) typically improve recall at the expense of precision, but are more readily retargetable for new nomenclatures and organisms. Hybrid systems (Tanabe and Wilbur, 2002; Mika and Rost, 2004) attempt to take advantage of both approaches. Although these approaches tend to generate acceptable recognition, they are heavily dependent on the type of data on which they are trained.

(Fukuda et al., 1998) proposed a rule-based protein name extraction system called PROPER (PROtein Proper-noun phrase Extracting Rules) system, which utilizes a set of rules based on the surface form of text in conjunction with a Part-Of-Speech (POS) tagging to identify what looks like a protein without referring to any specific BNE dictionary. They reported a 94.7% precision and a 98.84% recall for the identification of BNEs. The results that they achieved seem to be too specific to their training and test sets.

(Hanisch et al., 2003) proposed a rule-based protein and gene name extraction system called ProMiner, which is based on the construction of a general-purpose dictionary along with different dictionaries of synonyms and an automatic curation procedure based on a simple token model of protein names. Results showed that their system achieved a 0.80 F-measure score in the name extraction task on the BioCreative test set (BioCreative).

(Yamamoto et al., 2003) proposed the use of morphological analysis to improve protein name tagging. Their approach tags proteins based on morpheme chunking to properly determine protein name boundary. They used the GENIA corpus for training and testing and obtained an F-measure score of 0.70 for protein name tagging.

(Collier et al., 2000) used a machine learning approach to protein name extraction based on a linear interpolation Hidden Markov Model (HMM) trained using bi-grams. They focused on finding the most likely protein sequence classes (C) for a given sequence of words (W), by maximizing the probability of C given W, P(C—W). Unlike traditional dictionary based methods, the approach uses no manually crafted patterns. However, their approach may misidentify term boundaries for phrases containing potentially ambiguous local structures such as coordination and parenthesis. They reported an F-measure score of 0.73 for different mixtures of models tested on 20 abstracts.

(Kazama et al., 2002) proposed a machine learning approach to BNE tagging based on support vector machines (SVM), which was trained on the GENIA corpus. Their preliminary results of the system showed that the SVM with the polynomial kernel function outperforms techniques of Maximum Entropy based systems.

Yet another BNE tagging system is ABNER (Settles, 2005), which utilizes machine learning, namely conditional random fields, with a variation of orthographic and contextual features and no semantic or syntactic features. ABNER achieves an F-measure score of 0.71 on the NLPA 2004 shared task dataset corpus and 0.70 on the BioCreative corpus.and scored an F1-measure of 51.8set.

(Tanabe and Wilbur, 2002) used a combination of statistical and knowledge-based strategies, which utilized automatically generated rules from transformation based POS tagging and other generated rules from morphological clues, low frequency trigrams, and indicator terms. A key step in their method is the extraction of multi-word gene and protein names that are dominant in the corpus but inaccessible to the POS tagger. The advantage of such an approach is that it is independent of any biomedical domain. However, it can miss single word gene names that do not occur in contextual gene theme terms. It can also incorrectly tag compound gene names, plasmids, and phages.

(Mika and Rost, 2004) developed NLProt, which

combines the use of dictionaries, rules-based filtering, and machine learning based on an SVM classifier to tag protein names in MEDLINE. The NLProt system used rules for pre-filtering and the SVM for classification, and it achieved a precision of 75% and recall 76%.

2.2 Relationship Extraction

As for the extraction of interactions, most efforts in extraction of biomedical interactions between entities from text have focused on using rule-based approaches due to the familiarity of medical terms that tend to describe interactions. These approaches have proven to be successful with notably good results. In these approaches, most researchers attempted to define an accurate set of rules to describe relationship types and patterns and to build ontologies and dictionaries to be consulted in the extraction process. These rules, ontologies, and dictionaries are typically domain specific and are often not generalizable to other problems.

(Blaschke et al., 1999) reported a domain specific approach for extracting protein-protein interactions from biomedical text based on a set of predefined patterns and words describing interactions. Later work attempted to automatically extract interactions, which are referenced in the database of interacting proteins (Xenarios et al., 2000), from the text mentioning the interactions (Blaschke and Valencia, 2001). They achieved surprisingly low recall (25%), which they attributed to problems in properly identifying protein names in the text.

(Koike et al., 2005) developed a system called PRIME, which was used to extract biological functions of genes, proteins, and their families. Their system used a shallow parser and sentence structure analyzer. They extracted so-called ACTOR-OBJECT relationships from the shallow parsed sentences using rule based sentence structure analysis. The identification of BNEs was done by consulting the GENA gene name dictionary and family name dictionary. In extracting the biological functions of genes and proteins, their system reported a recall of 64% and a precision of 94%.

Saric et al. developed a system to extract gene expression regulatory information in yeast as well as other regulatory mechanisms such phosphorylation (Saric et al., 2004; Saric et al., 2006). They used a rule based named entity recognition module, which recognizes named entities via cascading finite state automata. They reported a precision of 83-90% and 86-95% for the extraction of gene expression and phosphorylation regulatory information respectively.

(Leroy and Chen, 2005) used linguistic parsers and Concept Spaces, which use a generic co-occurrence based technique that extracts relevant medical phrases using a noun chunker. Their system employed UMLS (Humphreys and Lindberg, 1993), GO (Ashburner et al., 2000), and GENA (Koike and Takagi, 2004) to further improve extraction. Their main purpose was entity identification and cross reference to other databases to obtain more knowledge about entities involved in the system.

Other extraction approaches such as the one reported on by (Cooper and Kershenbaum, 2005) utilized a large manually curated dictionary of many possible combinations of gene/protein names and aliases from different databases and ontologies. They annotated their corpus using a dictionary-based longest matching technique. In addition, they used filtering with a maximum entropy based named entity recognizer in order to remove the false positives that were generated from merging databases. The problem with this approach is the resulting inconsistencies from merging databases, which could hurt the effectiveness of the system. They reported a recall of 87.1 % and a precision of 78.5% in the relationship extraction task.

Work by (Mack et al., 2004) used the Munich Information Center for Protein Sequences (MIPS) for entity identification. Their system was integrated in the IBM Unstructured Information Management Architecture (UIMA) framework (Ferrucci and Lally, 2004) for tokenization, identification of entities, and extraction of relations. Their approach was based on a combination of computational linguistics, statistics, and domain specific rules to detect protein interactions. They reported a recall of 61% and a precision of 97%.

(Hao et al., 2005) developed an unsupervised approach, which also uses patterns that were deduced using minimum description lengths. They used pattern optimization techniques to enhance the patterns by introducing most common keywords that tend to describe interactions.

(Jörg et. al., 2005) developed Ali Baba which uses sequence alignments applied to sentences annotated with interactions and part of speech tags.It also uses finite state automata optimized with a genetic algorithm in its approach. It then matches the generated patterns against arbitrary text to extract interactions and their respective partners. The system scored an F1-measure of 51.8% on the LLL'05 evaluation set.

The aforementioned systems used either rule-based approaches, which require manual intervention from domain experts, or statistical approaches, either supervised or semi-supervised, which also require manually curated training data.

3 BioNoculars

BioNoculars is a relationship extraction system that based on a fully unsupervised technique suggested by (Hassan et al., 2006) to automatically extract protein-protein interaction from medical articles. It can be retargeted to different domains such as protein interactions in diseases. The only requirement is to compile domain specific taggers and dictionaries, which would aid the system in performing the required task.

The approach uses an unsupervised graph-based mutual reinforcement, which depends on the construction of generalized extraction patterns that could match instances of relationships (Hassan et al., 2006). Graph-based mutual reinforcement is similar to the idea of hubs and authorities in web pages depicted by the HITS algorithm (Kleinberg, 1998). The basic idea behind the algorithm is that the importance of a page increases when more and more good pages link to it. The duality between patterns and extracted information (tuples) leads to the fact that patterns could express different tuples, and tuples in turn could be expressed by different patterns. Tuple in this context contains three elements, namely two proteins and the type of interaction between them. The proposed approach is composed of two main steps, namely initial pattern construction and then pattern induction.

For pattern construction, the text is POS tagged and BNE tagged. The tags of Noun Phrases or sequences of nouns that constitute a BNE are removed and replaced with a BNE tag. Then, an n-gram language model is built on the tagged text (using tags only) and is used to construct weighted finite state machines. Paths with low cost (high language model probabilities) are chosen to construct the initial set of patterns; the intuition is that paths with low cost (high probability) are frequent and could represent potential candidate patterns. The number of candidate initial patterns could be reduced significantly by specifying the candidate types of entities of interest. In the case of BioNoculars, the focus was on relationships between BNEs of type PROTEIN. The candidate patterns are then applied to the tagged stream to produce in-sentence relationship tuples.

As for pattern induction, due to the duality in the patterns and tuples relation, patterns and tuples are represented by a bipartite graph as illustrated in Figure 1.

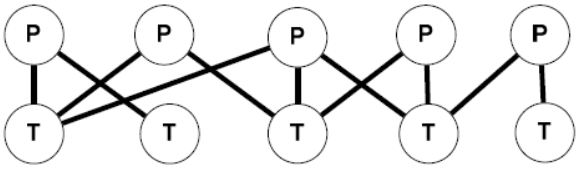


Figure 1: A bipartite graph representing patterns and tuples

Each pattern or tuple is represented by a node in the graph. Edges represent matching between patterns and tuples. The pattern induction problem can be formulated as follows: Given a very large set of data D containing a large set of patterns P, which match a large set of tuples T, the problem is to identify , which is the set of patterns that match the set of the most correct tuples *T*. The intuition is that the tuples matched by many different patterns tend to be correct and the patterns matching many different tuples tend to be good patterns. In other words, BioNoculars attempts to choose from the large space of patterns in the data the most informative, highest confidence patterns that could identify correct tuples; i.e. choosing the most authoritative patterns in analogy with the hub-authority problem. The most authoritative patterns can then be used for extracting relations from free text. The following pattern-tuple pairs show how patterns can match tuples in the corpus:

(protein) (verb) (noun) (prep.) (protein)

Cla4 induces phosphorylation of Cdc24

(protein) (I-protein) (Verb) (prep.) (protein)

NS5A interacts with Cdk1

The proposed approach represents an unsupervised technique for information extraction in general and particularly for relations extraction that requires no seed patterns or examples and achieves significant performance. Given enough domain text, the extracted patterns can support many types of sentences with different styles (such passive and active voice) and orderings (the interaction of X and Y vs. X interacts with Y).

One of the critical prerequisites of the above-mentioned approach is the use of a POS tagger, which is tuned for biomedical text, and a BNE tagger to properly identify BNEs. Both are critical for determining the types of relationships that are of interest. For POS tagging, a decision tree based tagger developed by (Schmid, 1994) was used in combination with a model, which was trained on a corrected/revised GENIA corpus provided by (Saric et al., 2004) and was reported to achieve 96.4% tagging accuracy (Saric et al., 2006). This POS tagger will be referred to as the Schmid tagger. For BNE tagging, ABNER was used. The accuracy of ABNER is approximately state of the art with precision and recall of 74.5% and 65.9% respectively with training done using the BioCreative corpora (BioCreative). Nonetheless we still face entity identification problems such as missed identifications in the text which in turn affects our results considerably. We do believe if we use a better identification method , we would yield better results.

4 Experimental Setup

Experiments aimed at extracting protein-protein interactions for Bakers yeast (Sacharomyces Cerevesiae) to assess BioNoculars (Cherry et al., 1998). The experiments were performed using 109,440 MEDLINE abstracts that contained the varying names of the yeast, namely Sacharomyces cerevisiae, S. Cerevisiae, Bakers yeast, Brewers yeast and Budding yeast. MEDLINE abstracts typically summarize the important aspects of papers possibly including protein-protein interactions if they are of relevance to the article. The goal was to deduce the most appropriate extraction patterns that can be later used to extract relations from any document. All the MEDLINE abstracts were used for pattern extraction except for 70 that were set aside for testing. There were no test documents in the training set. To build ground-truth, the test set was semi-manually POS and BNE tagged. They were also annotated with the interactions that are contained in the text. There was a condition that all the abstracts that are used for testing must have entries in the Database of Interacting Proteins and Protein-Protein Interactions (DIPPPI), which is a subset of the Database of Interacting Proteins (DIP) (Xenarios et al., 2000) restricted to proteins from yeast. DIPPPI lists the known protein-protein interactions in the MEDLINE abstracts. There were 297 protein-protein interactions in the test set of 70 abstracts. One of the disadvantages of DIPPPI is that the presence of interactions is indicated without mentioning their types or from which sentences they were extracted. Although BioNoculars is able to guess the sentence from which an interaction was extracted and the type of interaction, this information was ignored when evaluating against DIPPPI. Unfortunately, there is no standard test set for the proposed task, and most of the evaluation sets are proprietary. The authors hope that others can benefit from their test set, which is freely available.

The abstracts used for pattern extraction were POS tagged using the Schmid tagger and BNE tagging was done using ABNER. The patterns were restricted to only those with protein names. For extraction of interaction tuples, the test set was POS and BNE tagged using the Schmid tagger and ABNER respectively. A varying number of final patterns were then used to extract tuples from the test set and the average recall and precision were computed. Another setup was used in which the relationships were filtered using preset keywords for relationships such as inhibits, interacts, and activates to properly compare BioNoculars to systems in the literature that use such keywords. The keywords were obtained from the (Hakenberg et al., 2005) and (Temkin and Gilder, 2003). One of the generated pattern-tuple pairs was as follows:

(PROTEIN) (Verb) (Conjunction) (PROTEIN)

NS5A interacts with Cdk1

One consequence of tuple extraction is generation of redundant tuples, which contain the same enti-

Pattern Count	30	59	78	103	147	192	205	217
Recall	0.51	0.70	0.76	0.81	0.84	0.89	0.89	0.93
Precision	0.47	0.42	0.43	0.35	0.30	0.26	0.26	0.16
FMeasure	0.49	0.53	0.55	0.49	0.44	0.40	0.40	0.27

Table 1: Recall, Precision, and F-measure for extraction of tuples using a varying number of top rated patterns

Pattern Count	30	59	78	103	147	192	205	217
Recall	0.31	0.44	0.46	0.48	0.64	0.73	0.74	0.78
Precision	0.31	0.36	0.35	0.34	0.39	0.35	0.35	0.37
FMeasure	0.31	0.40	0.40	0.40	0.48	0.47	0.48	0.50

Table 2: Recall, Precision, and Recall for extraction of tuples using a varying number of top rated patters keyword filtering

ties and relations. Consequently, all protein aliases and full text names were resolved to a unified naming scheme and the unified scheme was used to replace all variations of protein names in patterns. All potential protein-protein interactions that BioNoculars extracted were compared to those in the DIPPPI databases.

5 Results and Discussion

For the first set of experiments, the experimental setup described above was used without modification. Table 1 and Figure 2 report on the resulting recall and precision when taking different number of highest rated patterns. The highest rated 217 patterns were divided on a linear scale into 8 clusters based on their relative weights.

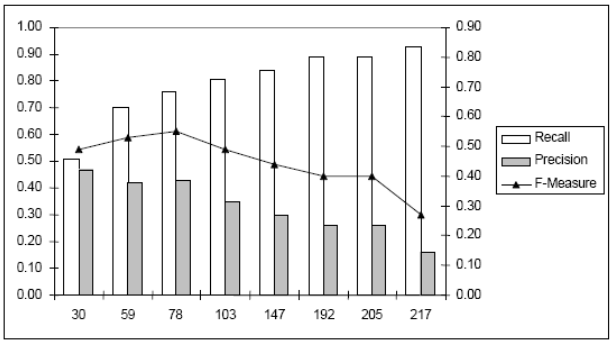


Figure 2: Recall, Precision, and F-measure for tuple extraction using a varying number of top patterns

As expected, Figure 2 clearly shows an inverse relationship between precision and recall. This is because using more extraction patterns yields more tuples thus increasing recall at the expense of precision. The F-measure (with $\beta = 1$) peeks at 78 patterns, which seems to provide the best score given that precision and recall are equally important. However, the technique seems to favor recall, reaching a recall of 93% when using all 217 patterns. The low precision levels warrant thorough investigation.

In the second set of experiments, extracted tuples were filtered using preset keywords indicating interactions. Table 2 and Figure 3 show the results of the experiments.

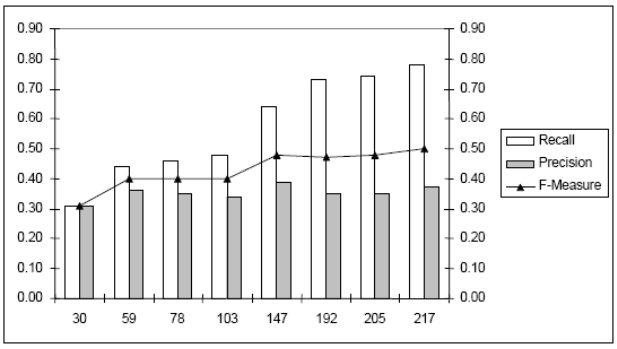


Figure 3: Recall, Precision, and F-measure for tuple extraction using a varying number of top patterns with keyword filtering

The results show that filtering with keywords led to lower recall, but precision remained fairly steady as the number of patterns changed. Nonetheless, the best precision in Figure 3 is lower than the best precision in Figure 2 and the maximum F-measure for this set of experiments is lower than the maximum F-measure when no filtering was used. The BioNoculars system with no filtering can be advantageous for recall oriented applications. The use of no filtering suggests that some interaction may be expressed in more generic forms or patterns. An intermediate solution would be to increase the size of the list of most commonly occurring keywords to filter the extracted tuples further.

Currently, ABNER, which is used by the system, has a precision of 75.4% and a recall of 65.9%. Perhaps improved tagging may improve the extraction effectiveness.

The effectiveness of BioNoculars needs to be

thoroughly compared to existing systems via the use of standard test sets, which are not readily available. Most of previously reported work has been tested on proprietary test sets or sets that are not publicly available. The creation of standard publicly available test set can prompt research in this area.

6 Conclusion and Future Work

This paper presented a system for extracting protein-protein interaction from biomedical text call BioNoculars. BioNoculars uses a statistical unsupervised learning algorithm, which is based on graph mutual reinforcement and data redundancy to extract extraction patterns. The system is recall oriented and is able to properly extract 93% of the interaction mentions from test MEDLINE abstracts. Nonetheless, the systems precision remains low. Precision can be enhanced by using keywords that describe interactions to filter to the resulting interaction, but this would be at the expense of recall.

As for future work, more attention should be focused on improving extraction patterns. Currently, the system focuses on extracting interactions between exactly two proteins. Some of the issues that need to be handled include complex relationship (X and Y interact with A and B), linguistic variability (passive vs. active voice; presence of superfluous words such as modifiers, adjectives, and prepositional phrases), protein lists (W interacts with X, Y, and Z), nested interactions (W, which interacts with X, also interacts with Y). Resolving these issues would require an investigation of how patterns can be generalized in automatic or semi-automatic ways. Further, the identification of proteins in the text requires greater attention. Also, the BioNoculars approach can be combined with other rule-based approaches to produce better results.

References

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. 2000. *Gene ontology: tool for the unification of biology.* Nature Genetics,volume 25 pp.25-29.

BioCreative. 2004. [Online].

Blaschke C., M. A. Andrade, C. Ouzounis, and A. Valencia. 1999. *Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions*. ISMB99, pp. 60-67.

Blaschke, C. and A. Valencia. 2001. *Can Bibliographic Pointers for Known Biological Protein Interactions as a Case Study.* Comparative and Functional Genomics,vol. 2: 196-206.

Cherry, J. M., C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. 1998. *SGD: Saccharomyces Genome Database.* Nucleic Acids Research, 26, 73-9.

Chun, H. W., Y. Tsuruka, J. D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. Tsujii. 2006. *Extraction of Gene-Disease Relations from MEDLINE Using Domain Dictionaries and Machine Learning.* Pacific Symposium on Biocomputing 11:4-15.

Collier, N., C. Nobata, and J. Tsujii. 2000. *Extracting the Names of Genes and Gene Products with a Hidden Markov Model.* COLING, 2000, pp. 201207.

Cooper, J. and A. Kershenbaum. 2005. *Discovery of protein-protein interactions using a combination of linguistic, statistical and graphical information.* BMC Bioinformatics.

DIPPPI http://www2.informatik.hu-berlin.de/ hakenber/corpora. 2006.

Ferrucci, D. and A. Lally. 2004. *UIMA: an architectural approach to unstructured information processing in the corporate research environment.* Natural Language Engineering 10, No. 3-4, 327-348.

Fukuda, K., T. Tsunoda, A. Tamura, and T. Takagi. 1998. *Toward information extraction: identifying protein names from biological papers.* PSB, pages 705716.

Hakenberg, J., C. Plake, U. Leser, H. Kirsch, and D. Rebholz-Schuhmann. 2005. *LLL'05 Challenge: Genic Interaction Extraction with Alignments and Finite State Automata.* Proc Learning Language in Logic Workshop (LLL'05) at ICML 2005, pp. 38-45. Bonn, Germany.

Hanisch, D., J. Fluck, HT. Mevissen, and R. Zimmer. 2003. *Playing biologys name game: identifying protein names in scientific text.* PSB, pages 403414.

Hao, Y., X. Zhu, M. Huang, and M. Li. 2005. *Discovering patterns to extract protein-protein interactions from the literature: Part II.* Bioinformatics, Vol. 00 no. 0 2005 pages 1-7.

Hassan, H., A. Hassan, and O. Emam. 2006. *Unsupervised Information Extraction Approach Using Graph Mutual Reinforcement.* Proceedings of Empirical Methods for Natural Language Processing (EMNLP).

Humphreys B. L. and D. A. B. Lindberg. 1993. *The UMLS project: making the conceptual connection between users and the information they need.* Bulletin of the Medical Library Association, 1993; 81(2): 170.

Jörg Hakenberg, Conrad Plake, Ulf Leser. 2005. *Genic Interaction Extraction with Alignments and Finite State Automata.* Proc Learning Language in Logic Workshop (LLL'05) at ICML 2005, pp. 38-45. Bonn, Germany (August 2005)

Kazama, J., T. Makino, Y. Ohta, and J. Tsujii. 2002. *Tuning Support Vector Machines for Biomedical Named Entity Recognition.* ACL Workshop on NLP in Biomedical Domain, pages 18.

Kleinberg, J. 1998. *Authoritative sources in a hyperlinked environment.* In Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms, pages 668-677, ACM Press, New York.

Koike A. and T. Takagi. 2004. *Gene/protein/family name recognition in biomedical literature.* BioLINK 2004: Linking Biological Literature, Ontologies, and Database, pp. 9-16.

Koike, A., Y. Niwa, and T. Takagi 2005. *Automatic extraction of gene/protein biological functions from biomedical text.* Bioinformatics, Vol. 21, No. 7.

Leroy, G. and H. Chen. 2005. *Genescene: An Ontology-enhanced Integration of Linguistic and Co-Occurance based Relations in Biomedical Text.* JASIST Special Issue on Bioinformatics.

Mack, R. L., S. Mukherjea, A. Soffer, N. Uramoto, E. W. Brown, A. Coden, J. W. Cooper, A. Inokuchi, B. Iyer, Y. Mass, H. Matsuzawa, L. V. Subramaniam. 2004. *Text analytics for life science using the Unstructured Information Management Architecture.* IBM Systems Journal 43(3): 490-515.

Mika, S. and B. Rost. 2004. *NLProt: extracting protein names and sequences from papers.* Nucleic Acids Research, 32 (Web Server issue): W634W637.

Saric, J., L. J. Jensen, R. Ouzounova, I. Rojas, and P. Bork. 2004. *Extracting regulatory gene expression networks from PUBMED.* Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, pp.191-198.

Saric, J., L. J. Jensen, R. Ouzounova, I. Rojas, and P. Bork. 2006. *Extraction of regulatory gene/protein networks from Medline.* Bioinformatics Vol.22 no 6,pp. 645-650.

Schmid, H. 1994. *Probabilistic Part-of-Speech Tagging Using Decision Trees.* In the International Conference on New Methods in Language Processing, Manchester, UK.

Settles, B. 2004. *Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets.* In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), Geneva, Switzerland, pages 104-107.

Settles, B. 2005. *ABNER: an open source tool for automatically tagging genes, proteins, and other entity names in text.* Bioinformatics, 21(14): 3191-3192.

Tanabe L., and W. J. Wilbur. 2002. *Tagging gene and protein names in biomedical text.* Bioinformatics, 18(8):11241132.

Temkin, J. M. and M. R. Gilder. 2003. *Extraction of protein interaction information from unstructured text using a context-free grammar.* Bioinformatics 19(16):2046-2053.

Xenarios I, Rice DW, Salwinski L, Baron MK, Marcotte EM, Eisenberg D. 2000. *DIP: the Database of Interacting Proteins.* Nucleic Acids Res 28: 289291.

Yamamoto, K., T. Kudo, A. Konagaya, Y. Matsumoto. 2003. *Protein Name Tagging for Biomedical Annotation in Text.* Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, pp. 65-72.

A Shared Task Involving Multi-label Classification of Clinical Free Text

John P. Pestian[1], **Christopher Brew**[2], **Paweł Matykiewicz**[1,4],
DJ Hovermale[2], **Neil Johnson**[1], **K. Bretonnel Cohen**[3],
Włodzisław Duch[4]
[1]Cincinnati Children's Hospital Medical Center, University of Cincinnati,
[2]Ohio State University, Department of Linguistics,
[3]University of Colorado School of Medicine,
[4]Nicolaus Copernicus University, Toruń, Poland.

Abstract

This paper reports on a shared task involving the assignment of ICD-9-CM codes to radiology reports. Two features distinguished this task from previous shared tasks in the biomedical domain. One is that it resulted in the first freely distributable corpus of fully anonymized clinical text. This resource is permanently available and will (we hope) facilitate future research. The other key feature of the task is that it required categorization with respect to a large and commercially significant set of labels. The number of participants was larger than in any previous biomedical challenge task. We describe the data production process and the evaluation measures, and give a preliminary analysis of the results. Many systems performed at levels approaching the inter-coder agreement, suggesting that human-like performance on this task is within the reach of currently available technologies.

1 Introduction

Clinical free text (primary data about patients, as opposed to journal articles) poses significant technical challenges for natural language processing (NLP). In addition, there are ethical and social demands when working with such data, which is intended for use by trained medical practitioners who appreciate the constraints that patient confidentiality imposes. State-of-the-art NLP systems handle carefully edited text better than fragmentary notes, and clinical language is known to exhibit unique sublanguage characteristics (Hirschman and Sager, 1982; Friedman et al., 2002; Stetson et al., 2002) (e.g. verbless sentences, domain-specific punctuation semantics, and unusual metonomies) that may limit the performance of general NLP tools. More importantly, the confidentiality requirements take time and effort to address, so it is not surprising that much work in the biomedical domain has focused on edited journal articles (and the genomics domain) rather than clinical free text in medical records. The fact remains, however, that the automation of healthcare workflows can bring important benefits to treatment (Hurtado et al., 2001) and reduce administrative burden, and that free text is a critical component of these workflows. There are economic motivations for the task, as well. The cost of adding labels like ICD-9-CM to clinical free text and the cost of repairing associated errors is approximately $25 billion per year in the US (Lang, 2007). For these (and many other) reasons, there have been consistent attempts to overcome the obstacles which hinder the processing of clinical text (Uzuner et al., 2006). This paper discusses one such attempt—The 2007 Computational Medicine Challenge, hereafter referred to as "the Challenge". There were two main reasons for conducting the Challenge. One is to facilitate advances in mining clinical free text; shared tasks in other biomedical domains have been shown to drive progress in the field in multiple ways (see (Hirschman and Blaschke, 2006; Hersh et al., 2005; Uzuner et al., 2006; Hersh et al., 2006) for a comprehensive review of biomedical challenge tasks and their contributions). The other is a ground-

BioNLP 2007: Biological, translational, and clinical language processing, pages 97–104,
Prague, June 2007.

breaking distribution of useful, reusable, carefully anonymized clinical data to the research community, whose data use agreement is simply to cite the source. The remaining sections of this paper describe how the data were prepared, the methods for scoring, preliminary results [to be updated if submission is accepted—results are currently still under analysis], and some lessons learned.

2 Corpus collection and coding process

Supervised methods for machine learning require training data. Yet, due to confidentiality requirements, spotty electronic availability, and variance in recording standards, the requisite clinical training data are difficult to obtain. One goal of the challenge was to create a publicly available ”gold standard” that could serve as the seed for a larger, open-source clinical corpus. For this we used the following guiding principles: individual identity must be expunged to meet United States HIPAA standards, (U.S. Health, 2002) and approved for release by the local Institutional Review Board (IRB); the sample must represent problems that medical records coders actually face; the sample must have enough data for machine-learning-based systems to do well; and the sample must include proportionate representations of very low-frequency classes.

Data for the corpus were collected from the Cincinnati Children's Hospital Medical Center's (CCHMC) Department of Radiology. CCHMC's Institutional Review Board approved release of the data. Sampling of all outpatient chest x-ray and renal procedures for a one-year period was done using a bootstrap method (Walters, 2004). These data are among those most commonly used, and are designed to provide enough codes to cover a substantial proportion of pediatric radiology activity. Expunging patient identity to meet HIPAA standards included three steps: disambiguation, anonymization, and data scrubbing (Pestian et al., 2005).

Ambiguity and Anonymization. Not surprisingly, some degree of disambiguation is needed to carry out effective anonymization (Uzuner et al., 2006; Sibanda and Uzuner, 2006). The reason is that clinical text is dense with medical jargon, abbreviations, and acronyms, many of which turn out to be ambiguous between a sense that needs anonymization and a different sense that does not. For example, in a clinical setting, *FT* can be an abbreviation for *full-term*, *fort* (as in *Fort Bragg*), *feet*, *foot*, *field test*, *full-time* or *family therapy*. *Fort Bragg*, being a place name, and a possible component of an address, could indirectly lead to identification of the patient. Until such occurrences are disambiguated, it is not possible to be certain that other steps to anonymize data are adequate. To resolve the relevant ambiguities found in this free text, we relied on previous efforts that used expert input to develop clinical disambiguation rules (Pestian et al., 2004).

Anonymization. To assure patient privacy, clinical text that is used for non-clinical reasons must be anonymized. However, to be maximally useful for machine-learning, this must be done in a particular way. Replacing personal names with some unspecific value such as ”*” would lose potentially useful information. Our goal is to replace the sensitive fields with *like* values that obscure the identity of the individual (Cho et al., 2002). We found that the amount of sensitive information routinely found in unstructured free text data is limited. In our case, these data included patient and physician names and sometimes dates or geographic locations, but little or no other sensitive information turned up in the relevant database fields. Using our internally developed encryption broker software, we replaced all female names with “Jane”, all male names with ”John”, and all surnames with ”Johnson”. Dates were randomly shifted.

Manual Inspection. Once the data were disambiguated and anonymized, they were manually reviewed for the presence of any Protected Health Information (PHI). If a specific token was perceived to potentially violate PHI regulations, the entire record was deleted from the dataset. In some case, however, a general geographic area was changed and not deleted. For example if the data read ”patient lived near Mr. Roger's neighborhood” it would be deleted, because it may be traceable. On the other hand, if the data read ”patient was from Cincinnati” it may have been changed to read ”patient was from the Covington” After this process, a corpus of 2,216 records was obtained (See Table 2 for details).

ICD-9-CM Assignment. A radiology report has multiple components. Two parts in particular are essential for the assignment of ICD-9-CM codes:

clinical history—provided by an ordering physician before a radiological procedure, and *impression*—reported by a radiologist after the procedure. In the case of radiology reports, ICD-9-CM codes serve as justification to have a certain procedure performed. There are official guidelines for radiology ICD-9-CM coding (Moisio, 2000). These guidelines note that every disease code requires a minimum number of digits before reimbursement will occur; that a definite diagnosis should always be coded when possible; that an uncertain diagnosis should never be coded; and that symptoms must never be coded when a definite diagnosis is available. Particular hospitals and insurance companies typically augment these principles with more specific internal guidelines and practices for coding. For these reasons of policy, and because of natural variation in human judgment, it is not uncommon for multiple annotators to assign different codes to the same text. Understanding the sources of this variation is important; so too is the need to create a definite gold standard for use in the challenge. To do so, data were annotated by the coding staff of CCHMC and two independent coding companies: COMPANY Y and COMPANY Z.

Majority annotation. A single gold standard was created from these three sets of annotations. There was no reason to adopt any *a priori* preference for one annotator over another, so the democratic principle of assigning a majority annotation was used. The majority annotation consists of those codes assigned to the document by two or more of the annotators. There are, however, several possible problems with this approach. For example, it could be that the majority annotation will be empty. This will be rare (126 records out of 2,216 in our case), because it only happens when the codes assigned by the three annotators form disjoint sets. In most hospital systems, including our own, the coders are required to indicate a primary code. By convention, the primary code is listed as the record's first code, and has an especially strong impact on the billing process. For simplicity's sake, the majority annotation process ignores the distinction between primary and secondary codes. There is space for a better solution here, but we have not seriously explored it. We have, however, conducted an analysis of agreement statistics (not further discussed here) that suggests that the overall effect of the majority method is to create a coding that shares many statistical properties with the originals, except that it reduces the effect of the annotators' individual idiosyncrasies. The majority annotation is illustrated in Table 1.

Our evaluation strategy makes the simplistic assumption that the majority annotation is a true gold standard and a worthwhile target for emulation. This is debatable, and is discussed below, but for the sake of definiteness we simply stipulate that submissions will be compared against the majority annotation, and that the best possible performance is to exactly replicate said majority annotation.

3 Evaluation

Micro- and macro-averaging. Although we rank systems for purposes of determining the top three performers on the basis of micro-averaged F1, we report a variety of performance data, including the micro-average, macro-average, and a cost-sensitive measure of loss. Jackson and Moulinier comment (for general text classification) that: "No agreement has been reached...on whether one should prefer micro- or macro-averages in reporting results. Macro-averaging may be preferred if a classification system is required to perform consistently across all classes regardless of how densely populated these are. On the other hand, micro-averaging may be preferred if the density of a class reflects its importance in the end-user system" (Jackson and Moulinier, 2002):160-161. For the present medical application, we are more interested in the number of patients whose cases are correctly documented and billed than in ensuring good coverage over the full range of diagnostic codes. We therefore emphasize the micro-average.

A cost-sensitive accuracy measure. While F-measure is well-established as a method for ranking, there are reasons for wanting to augment this with a cost-sensitive measure. An approach that allows penalties for over-coding (a false positive) and under-coding (a false negative) to be manipulated has important implications. The penalty for under-coding is simple—the hospital loses the amount of revenue that it would have earned if it had assigned the code. The regulations under which coding is done enforce an automatic over-coding penalty of

Table 1: Majority Annotation

	Hospital	Company Y	Company Z	Majority
Document 1	AB	BC	AB	AB
Document 2	BC	ABD	CDE	BCD
Document 3	EF	EF	E	EF
Document 4	ABEF	ACEF	CDEF	ACEF

three times what is earned from the erroneous code, with the additional risk of possible prosecution for fraud. This motivates a generalized version of Jaccard's similarity metric (Gower and Legendre, 1986), which was introduced by Boutell, Shen, Luo and Brown (Boutell et al., 2003).

Suppose that Y_x is the set of correct labels for a test set and P_x is the set of labels predicted by some participating system. Define $F_x = P_x - Y_x$ and $M_x = Y_x - P_x$, i.e. F_x is the set of false positives, and M_x is the set of missed labels or false negatives. The score is given by

$$score(P_x) = \left(1 - \frac{\beta|M_x| + \gamma|F_x|}{|Y_x \cup P_x|}\right)^{\alpha} \quad (1)$$

As noted in (Boutell et al., 2003), if $\beta = \gamma = 1$ this formula reduces to the simpler case of

$$score(P_x) = \left(1 - \frac{|Y_x \cap P_x|}{|Y_x \cup P_x|}\right)^{\alpha} \quad (2)$$

The discussion in (Boutell et al., 2003) points out that constraints are necessary on β and γ to ensure that the inner term of the expression is non-negative. We do not understand the way that they formulate these constraints, but note that non-negativity will be ensured if $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$. Since overcoding is three times as bad as undercoding, we use $\gamma = 1.0$, $\beta = 0.33$. Varying the value of α would affect the range of the scores, but does not alter the rankings of individual systems. We therefore used $\alpha = 1$. This measure does not represent the possibility of prosecution for fraud, because the costs involved are incommensurate with the ones that are represented. With these parameter settings, the cost-sensitive measure produces rankings that differ considerably from those produced by macro-averaged balanced F-measure. For example, we shall see that the system ranked third in the competition by macro-averaged F-measure assigns a total of 1167 labels, where the second-ranked assigns 1232, and the cost-sensitive measure rewards this conservatism in assigning labels by reversing the ranking of the two systems. In either case, the difference between the systems is small (0.86% difference in F-measure, 0.53% difference in the cost-sensitive measure).

4 The Data

We selected for the challenge a subset of the comprehensive data set described above. The subset was created by stratified sampling, such that it contains 20% of the documents in each category. Thus, the proportion of categories in the sample is the same as the proportion of categories in the full data set. We included in the initial sample only those categories to which 100 or more documents from the comprehensive data set were assigned. After the process summarized in Table 2, the data were divided into two partitions: a training set with 978 documents, and a testing set with 976. Forty-five ICD-9-CM labels (e.g 780.6) are observed in these data sets. These labels form 94 distinct combinations (e.g. the combination 780.6, 786.2). We required that any combination have at least two exemplars in the data, and we split each combination between the training and the test sets. So, there may be labels and combinations of labels that occur only one time in the training data, but participants can be sure that no combination will occur in the test data that has not previously occurred at least once in the training data. Our policy here has the unintended consequence that any combination that appears exactly once in the training data is highly likely to appear exactly once in the test data. This gives unnecessary information to the participants. In future challenges we will drop the requirement for two occurrences in the data, but ensure that single-occurrence combinations are allocated to the training set rather than the

test set. This maintains the guarantee that there will be no unseen combinations in the test data. The full data set may be downloaded from the official challenge web-site.

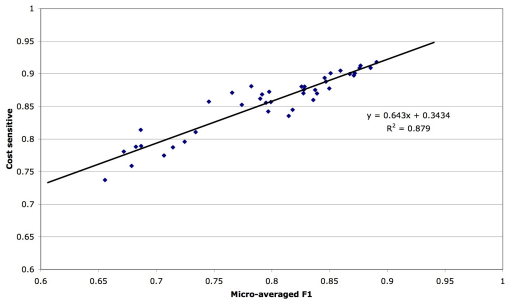
5 Results

Notice of the Challenge was distributed using electronic mailing lists supplied by the Association of Computational Linguistics, IEEE Computer Intelligence and Data Mining, and American Medical Informatics Association's Natural Language Processing special interest group. Interested participants were asked to register at the official challenge website. Registration began February 1, 2007 and ended February 28, 2007. Approximately 150 individuals registered from 22 countries and six continents. Upon completing registration, an automated e-mail was sent with the location of the training data. On March 1, 2007 participants received notice of the location of the testing data. Participants were encouraged to use the data for other purposes as long as it was non-commercial and the appropriate citation was made. There were no other data use restrictions. Participants had until March 18, 2007 to submit their results and an explanation of their model. Approximately 33% (50) of the participants submitted results. During the course of the Challenge participants asked a range of questions. These were posted to the official challenge web-site - www.computationalmedicine.org/challenge.

The figure below is a scatterplot relating micro-averaged F1 to the cost-sensitive measure described above. Each point represents a system. The top-performing systems achieved 0.8908, the minimum was 0.1541, and the mean was 0.7670, with a SD of 0.1340. There are 21 systems with a micro-averaged F1 between 0.81 and 0.90. Another 14 have $F1 > 0.70$. It is noticeable that the systems are not ranked identically by the cost-sensitive and the micro-averaged measure, but the differences are small in each case.

A preliminary screening using a two-factor ANOVA with system identity and diagnostic code as predictive factors for balanced F-measure revealed a significant main effect of both system and code. Pairwise t-tests using Holm's correction for multiple comparisons revealed no statistically significant differences between the systems performing at F=0.70 or higher. Differences between the top system and a system with a microaveraged F-measure of 0.66 do come out significant on this measure.

Figure 1: Scatter plot of evaluation measures



We have also calculated (Table 3) the agreement figures for the three individual annotations that went into the majority gold standard. We see that CCHMC outranks COMPANY Y on the cost-sensitive measure, but the reverse is true for micro- and macro-averaged F1, with the agreement between the hospital and the gold standard being especially low for the macro-averaged version. To understand these figures, it is necessary to recall that the gold standard is a majority annotation that is formed from the the three component annotations. It appears that for rare codes, which have a disproportionate effect on the macro-averaged F, the majority annotation is dominated by cases where company Y and company Z assign the same code, one that CCHMC did not assign.

The agreement figures are comparable to those of the best automatic systems. If submitted to the competition, the components of the majority annotation would not have outranked the best systems, even though the components contributed to the majority opinion. It is tempting to conclude that the automated systems are close to human-level performance. Recall, however, that while the hospital and the companies did not have the luxury of exposure to the majority annotation, the systems did have that access, which allowed them to explicitly model the properties of that majority annotation. A more moderate conclusion is that the hospital and the companies might be able to improve (or at least adjust) their annotation practices by studying the majority

Table 2: Characteristics of the data set through the development process.

Step	Removed	Total documents
One-year collection of documents		20,275
20 percent sample of one-year collection		4,055
Manual inspection for anonymization problems	1,839	2,216
Removal of records with no majority code	126	2,090
Removal of records with a code occurring only once	136	1,954

Table 3: Comparison of human annotators against majority.

Annotator	Cost-sensitive	Micro-averaged F1	Macro-averaged F1
HOSPITAL	0.9056	0.8264	0.6124
COMPANY Y	0.8997	0.8963	0.8973
COMPANY Z	0.8621	0.8454	0.8829

annotation and adapting as appropriate.

6 Discussion

Compared to other recent text classification shared tasks in the biomedical domain (Uzuner et al., 2006; Hersh et al., 2004; Hersh et al., 2005), this task required categorization with respect to a set of labels more than an order of magnitude larger than previous evaluations. This increase in the size of the set of labels is an important step forward for the field–systems that perform well on smaller sets of categories do not necessarily perform well with larger sets of categories (Jackson and Moulinier, 2002), so the data set will allow for more thorough text categorization system evaluations than have been possible in the past. Another important contribution of the work reported here may be the distribution of the data—the first fully distributable, freely usable data set of clinical text. The high number of participants and final submissions was a pleasant surprise; we attribute this, among other things, to the fact that this was an applied challenge, that real data were supplied, and that participants were free to use these data in other venues.

Participants utilized a diverse range of approaches. These system descriptions are based on brief comments entered into the submission box, and are obviously subject to revision. The three highest scorers all mentioned "negation," all seemed to be using the structure of UMLS in a serious way. The better systems frequently mentioned "hypernyms" or "synonyms," and were apparently doing significant amounts of symbolic processing. Two of the top three had machine-learning components, while one of the top three used purely symbolic methods. The most common approach seems to be thoughtful and medically-informed feature engineering followed by some variety of machine learning. The top-performing system used C4.5, suggesting that use of the latest algorithms is not a pre-requisite for success. SVMs and related large-margin approaches to machine learning were strongly represented, but did not seem to be reliably predictive of high ranking.

6.1 Observations on running the task and the evaluation

The most frequently viewed question of the FAQ was related to a script to calculate the evaluation score. This was supplied both as a downloadable script and as an interactive web-page with a form for submission. In retrospect, we realize that we had not fully thought through what would happen as people began to use this script. If we run a similar contest in the future, we will be better prepared for the confusion that this can cause.

A novel aspect of this task was that although we only scored a single run on the test data, we allowed participants to submit their "final" run up to 10 times, and to see their score each time. Note that although

participants could see how their score varied on successive submissions, they did *not* have access to the actual test data or to the correct answers, and so there were no opportunities for special-purpose hacks to handle special cases in the test data. The average participant tried 5.27 (SD 3.17) submissions against the test data. About halfway through the submission period we began to realize that in a competitive situation, there are risks in providing the type of feedback given on the submission form. In future challenges, we will be judicious in selecting the number of attempts allowed and the provision of any type of feedback. As far as we can tell our general assumption that the scientific integrity of the participants was greater than the need to game the system is true. It is good policy for those administering the contest, however, to keep temptations to a minimum. Our current preference would be to provide only the web-page interface with no more than five attempts, and to tell participants only whether their submission had been accepted, and if so, how many items and how many codes were recognized.

We provided an XML schema as a precise and publicly visible description of the submission format. Although we should not have been, we were surprised when changes to the schema were required in order to accommodate small but unexpected variations in participant submissions. An even simpler submission format would have been good. The advantage of the approach that we took was that XML validation gave us a degree of sanity-checking at little cost. The disadvantage was that some of the necessary sanity-checking went beyond what we could see how to do in a schema.

The fact that numerous participants generated systems with high performance indicates that the task was reasonable, and that sufficient information about the coding task was either provided by us or inferred by the participants to allow them to do their work. Since this is a first attempt, it is not yet clear what the upper limits on performance are for this task, but preliminary indications are that automated systems are or will soon be viable as a component of deployed systems for this kind of application.

7 Acknowledgements

The authors thank Aaron Cohen of the Oregon Health and Science University for observations on the inter-rater agreement between the three sources and its relationship to the majority assignments, and also for his input on testing for statistically significant differences between systems. We also thank PERSON of ORGANIZATION for helpful comments on the manuscript. Most importantly we thank all the participants for their on-going commitment, professional feedback and scientific integrity.

References

[Boutell et al., 2003] Boutell M., Shen X., Luo J. and Brown C. 2003. *Multi-label Semantic Scene Classification*, Technical Report 813. Department of Computer Science, University of Rochester September.

[Cho et al., 2002] Cho P. S., Taira R. K., and Kangarloo H. 2002 Text boundary detection of medical reports. *Proceedings of the Annual Symposium of the American Medical Informatics Association*, 998.

[Friedman et al., 2002] Friedman C., Kra P., and Rzhetsky A. 2002. Two biomedical sublanguages: a description based on the theories of Zellig Harris. *Journal of Biomedical Informatics*, 35:222–235.

[Gower and Legendre, 1986] Gower J. C. and Legendre P. 1986. Metric and euclidean properties of dissimilarity coefficient. *Journal of Classification*, 3:5–48.

[Hersh et al., 2004] Hersh W., Bhupatiraju R. T., Ross L., Roberts P., Cohen A. M., and Kraemer D. F. 2004. TREC 2004 Genomics track overview. *Proceedings of the 13th Annual Text Retrieval Conference*. National Institute of Standards and Technology.

[Hersh et al., 2006] Hersh W., Cohen A. M., Roberts P., and Rekapalli H. K. 2006. TREC 2006 Genomics track overview. *Proceedings of the 15th Annual Text Retrieval Conference* National Institute of Standards and Technology.

[Hersh et al., 2005] Hersh W., Cohen A. M., Yang J., Bhupatiraju R. T., Roberts P., and Hearst M. 2005. TREC 2005 Genomics track overview. *Proceedings of the 14th Annual Text Retrieval Conference*. National Institute of Standards and Technology.

[Hirschman and Blaschke, 2006] Hirschman L. and Blaschke C. 2006. Evaluation of text mining in biology. *Text mining for biology and biomedicine*, Chapter 9. Ananiadou S. and McNaught J., editors. Artech House.

[Hirschman and Sager, 1982] Hirschman L. and Sager S. 1982. Automatic information formatting of a medical sublanguage. *Sublanguage: studies of language in restricted semantic domains*, Chapter 2. Kittredge R. and Lehrberger J., editors. Walter de Gruyter.

[Hurtado et al., 2001] Hurtado M. P, Swift E. K., and Corrigan J. M. 2001. Crossing the Quality Chasm: A New Health System for the 21st Century. Institute of Medicine, National Academy of Sciences.

[Jackson and Moulinier, 2002] Jackson P. and Moulinier I. 2002. *Natural language processing for online applications: text retrieval, extraction, and categorization*. John Benjamins Publishing Co.

[Lang, 2007] Lang, D. 2007. CONSULTANT REPORT - Natural Language Processing in the Health Care Industry. Cincinnati Children's Hospital Medical Center, Winter 2007.

[Moisio, 2000] Moisio M. 2000. *A Guide to Health Care Insurance Billing*. Thomson Delmar Learning, Clifton Park.

[Pestian et al., 2005] Pestian J. P., Itert L., Andersen C. L., and Duch W. 2005. Preparing Clinical Text for Use in Biomedical Research. *Journal of Database Management*, 17(2):1-12.

[Pestian et al., 2004] Pestian J. P., Itert L., and Duch W. 2004. Development of a Pediatric Text-Corpus for Part-of-Speech Tagging. *Intelligent Information Processing and Web Mining, Advances in Soft Computing*, 219–226 New York, Springer Verlag.

[Sammuelsson and Wiren, 2000] Sammuelsson C. and Wiren M. 2000. Parsing Techniques. *Handbook of Natural Language Processing*, 59–93. Dale R., Moisl H., Somers H., editors. New York, Marcel Deker.

[Sibanda and Uzuner, 2006] Sibanda T. and Uzuner O. 2006. Role of local context in automatic deidentification of ungrammatical, fragmented text. *Proceedings of the Human Language Technology conference of the North American chapter of the Association for Computational Linguistics*, 65–73.

[Stetson et al., 2002] Stetson P. D., Johnson S. B., Scotch M., and Hripcsak G. 2002. The sublanguage of cross-coverage. *Proceedings of the Annual Symposium of the American Medical Informatics Association*, 742–746.

[U.S. Health, 2002] U.S. Heath & Human Services. 2002. 45 CFR Parts 160 and 164 Standards for Privacy of Individually Identifiable Health Information *Final Rule Federal Register*, 67(157):53181–53273.

[Uzuner et al., 2006] Uzuner O., Szolovits P., and Kohane I. 2006. i2b2 workshop on natural language processing challenges for clinical records. *Proceedings of the Fall Symposium of the American Medical Informatics Association*.

[Walters, 2004] Walters S. J. 2004. Sample size and power estimation for studies with health related quality of life outcomes: a comparison of four methods using the SF-36 *Health and Quality of Life Outcomes*, 2:26.

From Indexing the Biomedical Literature to Coding Clinical Text: Experience with MTI and Machine Learning Approaches

Alan R. Aronson[1], **Olivier Bodenreider**[1], **Dina Demner-Fushman**[1], **Kin Wah Fung**[1], **Vivian K. Lee**[1,2], **James G. Mork**[1], **Aurélie Névéol**[1], **Lee Peters**[1], **Willie J. Rogers**[1]

[1]Lister Hill Center
National Library of Medicine
Bethesda, MD 20894
{alan, olivier, demnerd, kwfung, mork, neveola, peters, wrogers}@nlm.nih.gov

[2]Vanderbilt University
Nashville, TN 37235
vivian.lee@vanderbilt.edu

Abstract

This paper describes the application of an ensemble of indexing and classification systems, which have been shown to be successful in information retrieval and classification of medical literature, to a new task of assigning ICD-9-CM codes to the clinical history and impression sections of radiology reports. The basic methods used are: a modification of the NLM Medical Text Indexer system, SVM, k-NN and a simple pattern-matching method. The basic methods are combined using a variant of stacking. Evaluated in the context of a Medical NLP Challenge, fusion produced an F-score of 0.85 on the Challenge test set, which is considerably above the mean Challenge F-score of 0.77 for 44 participating groups.

1 Introduction

Researchers at the National Library of Medicine (NLM) have developed the Medical Text Indexer (MTI) for the automatic indexing of the biomedical literature (Aronson et al., 2004). The unsupervised methods within MTI were later successfully combined with machine learning techniques and applied to the classification tasks in the Genomics Track evaluations at the Text Retrieval Conference (TREC) (Aronson et al., 2005 and Demner-Fushman et al., 2006). This fusion approach consists of using several basic classification methods with complementary strengths, combining the results using a modified ensemble method based on stacking (Ting and Witten, 1997).

While these methods have shown reasonable performance on indexing and retrieval tasks of biomedical articles, it remains to be determined how they would perform on a different biomedical corpus (e.g., clinical text) and on a different task (e.g., coding to a different controlled vocabulary). However, except for competitive evaluations such as TREC or BioCreAtIvE, corpora and gold standards for such tasks are generally not available, which is a limiting factor for such studies. For a survey of currently available corpora and developments in biomedical language processing, see Hunter and Cohen, 2006.

The Medical NLP Challenge[1] sponsored by a number of groups including the Computational Medicine Center (CMC) at the Cincinnati Children's Hospital Medical Center gave us the opportunity to apply our fusion approach to a clinical corpus. The Challenge was to assign ICD-9-CM codes (International Classification of Diseases, 9th Revision, Clinical Modification)[2] to clinical text consisting of anonymized clinical history and impression sections of radiology reports.

The Medical NLP Challenge organizers distributed a training corpus of almost 1,000 of the anonymized, abbreviated radiology reports along with

[1] See www.computationalmedicine.org/challenge/.
[2] See www.cdc.gov/nchs/icd9.htm.

BioNLP 2007: Biological, translational, and clinical language processing, pages 105–112,
Prague, June 2007.

gold standard ICD-9-CM assignments for each report obtained via a consensus of three independent sets of assignments. The primary measure for the Challenge was defined as the balanced F-score, with a secondary measure being cost-sensitive accuracy. These measures were computed for submissions to the Challenge based on a test corpus similar in size to the training corpus but distributed without gold standard code assignments.

The main objective of this study is to determine what adaptation of the original methods is required to code clinical text with ICD-9-CM, in contrast to indexing and retrieving MEDLINE®. Note that an earlier study (Gay et al., 2005) showed that only minor adaptations were required in extending the original model to full-text biomedical articles. A secondary objective is to evaluate the performance of our methods in this new setting.

2 Methods

In early experimentation with the training corpus provided by the Challenge organizers, we discovered that several of the training cases involved negated assertions in the text and that deleting these improved the performance of all basic methods being tested. For example, "no pneumonia" occurs many times in the impression section of a report, sometimes with additional context. Section 2.1 describes the process we used to remove these negated expressions; section 2.2 consists of descriptions of the four basic methods used in this study; and section 2.3 defines the fusion of the basic methods to form a final result.

2.1 Document Preparation

The NegEx program (Chapman et al., 2001a and 2001b, and Goldin and Chapman, 2003), which discovers negated expressions in text, was used to find negated expressions in the training and test corpora using a dictionary generated from concepts from the 2006AD version of the UMLS® Metathesaurus® (excluding the AMA vocabularies). A table containing the concept unique identifier (CUI) and English string (STR with LAT='ENG') was extracted from the main concept table, MRCON, and was used as input to NegEx to generate a dictionary that was later used as the universe of expressions which NegEx could find to be negated in the target corpora. (See the Appendix for examples of the input and output to this process.)

The XML text of the training and test corpora was converted to a tree representation and then traversed, operating on one radiology report at a time. The clinical history and impression sections of each report were tokenized to allow whitespace to be separated from the punctuation, numbers and alphabetic text. The concepts from the UMLS were tokenized in the same way, to allow the concepts found by NegEx to be aligned with the text. The negation phrases discovered by NegEx were also tokenized to find the appropriate negation phrase preceding or trailing the target concept. Using the location information obtained by matching the set of one or more target concepts and the associated negation phrase, the overlapping concept spans were merged and the span for the negation phrase and the outermost negated concept was removed. Any intervening concepts associated with the same negation phrase were removed, too. The abbreviated tree representation was then re-serialized back into XML.

As an example of our use of NegEx, consider the report with clinical history "13-year 2-month - old female evaluate for cough." and impression "No focal pneumonia." After removal of negated text, the clinical history becomes "13-year 2-month - old female", and the discussion is empty.

2.2 Basic Methods

The four basic methods used for the Medical NLP Challenge are MTI (a modification of NLM's Medical Text Indexer system), SVM (Support Vector Machines), k-NN (k Nearest Neighbors) and Pattern Matching (a simple, pattern-based classifier). Each of these methods is described here. Note that the MTI method uses a "Restrict to ICD-9-CM" algorithm that is described in the next section.

MTI. The original Medical Text Indexer (MTI) system, shown in Figure 1, consists of an infrastructure for applying alternative methods of discovering MeSH® headings for citation titles and abstracts and then combining them into an ordered list of recommended indexing terms. The top portion of the diagram consists of two paths, or methods, for creating a list of recommended indexing terms: MetaMap Indexing and PubMed® Related Citations. The MetaMap Indexing path actually

computes UMLS Metathesaurus concepts, which are passed to the Restrict to MeSH process (Bodenreider et al., 1998). The results from each path are weighted and combined using Post-Processing, which also refines the results to conform to NLM indexing policy. The system is highly parameterized not only by path weights but also by several parameters specific to the Restrict to MeSH and Post-Processing processes.

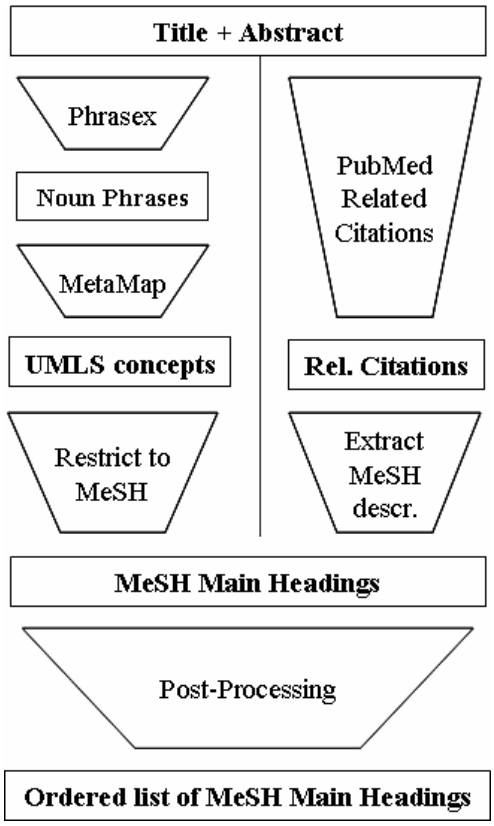


Figure 1: Medical Text Indexer (MTI) System

For use in the Challenge, the Medical Text Indexer (MTI) program itself required few adaptations. Most of the changes involved the environment from which MTI obtains the data it uses without changing the normal parameter settings. We also added a further post-processing component to filter our results.

For the environment, we replaced MTI's normal "Restrict to MeSH" algorithm with a "Restrict to ICD-9-CM" algorithm, described below, in order to map UMLS concepts to ICD-9-CM codes instead of MeSH headings. We also trained the PubMed Related Citations component, TexTool (Tanabe and Wilbur, 2002), on the Medical NLP Challenge training data instead of the entire MEDLINE/PubMed database as is the case for normal MTI use at NLM. For both of these methods, we used the actual ICD-9-CM codes to mimic UMLS CUIs used internally by MTI.

To create the new training data for the TexTool (Related Citations), we reformatted the Medical NLP Challenge training data into a pseudo-MEDLINE format using the "doc id" component as the PMID, the "CLINICAL_HISTORY" text component for the Title, the "IMPRESSION" text component for the Abstract, and all of the "CMC_MAJORITY" codes as MeSH Headings (see Figure 2). This provided us with direct ICD-9-CM codes to work with instead of MeSH Headings.

```
<doc id="97663756" type="RADIOLOGY_REPORT">
  <codes>
    <code origin="CMC_MAJORITY" type="ICD-9-
CM">780.6</code>
    <code origin="CMC_MAJORITY" type="ICD-9-
CM">786.2</code>
    <code origin="COMPANY3" type="ICD-9-
CM">786.2</code>
    <code origin="COMPANY1" type="ICD-9-
CM">780.6</code>
    <code origin="COMPANY1" type="ICD-9-
CM">786.2</code>
    <code origin="COMPANY2" type="ICD-9-
CM">780.6</code>
    <code origin="COMPANY2" type="ICD-9-
CM">786.2</code>
  </codes>
  <texts>
    <text origin="CCHMC_RADIOLOGY"
type="CLINICAL_HISTORY">Cough and fever.</text>
    <text origin="CCHMC_RADIOLOGY"
type="IMPRESSION">Normal radiographic appear-
ance of the chest, no pneumonia.</text>
  </texts>
</doc>
```

```
PMID- 97663756
TI  - Cough and fever.
AB  - Normal radiographic appearance of the
chest, no pneumonia.
MH  - Fever (780.6)
MH  - Cough (786.2)
```

Figure 2: XML Medical NLP Training Data modified to pseudo-ASCII MEDLINE format

Within MTI we also utilized an experimental option for MetaMap (Composite Phrases), which provides a longer UMLS concept match than usual. We did not use the following: (1) UMLS concept-specific checking and exclusion sections; and (2) the MeSH Subheading generation, checking, and removal elements, since they were not needed for this Challenge. We then had MTI use the new Re-

strict to ICD-9-CM file and the new TexTool to generate its results.

Restrict to ICD-9-CM. The mapping of every UMLS concept to ICD-9-CM developed for the Medical NLP Challenge is an adaptation of the original mapping to MeSH, later generalized to any target vocabulary (Fung and Bodenreider, 2005). Based on the UMLS Metathesaurus, the mapping utilizes four increasingly aggressive techniques: synonymy, built-in mappings, hierarchical mappings and associative mappings. In order to comply with coding rules in ICD-9-CM, mappings to non-leaf codes are later resolved into leaf codes.

Mappings to ICD-9-CM are identified through ***synonymy*** when names from ICD-9-CM are included in the UMLS concept identified by MetaMap. For example, the ICD-9-CM code 592.0 *Calculus of kidney* is associated with the UMLS concept C0392525 *Nephrolithiasis* through synonymy.

Built-in mappings are mapping relations between UMLS concepts implied from mappings provided by source vocabularies in the UMLS. For example, the UMLS concept C0239937 *Microscopic hematuria* is mapped to the concept C0018965 (which contains the ICD-9-CM code 599.7 *Hematuria*) through a mapping provided by SNOMED CT.

In the absence of a mapping through synonymy or built-in mapping, a ***hierarchical mapping*** is attempted. Starting from the concept identified by MetaMap, a graph of ancestors is built by first using its parent concepts and broader concepts, then adding the parent concepts and broader concepts of each concept, recursively. Semantic constraints (based on semantic types) are applied in order to prevent semantic drift. Ancestor concepts closest to the MetaMap source concept are selected from the graph. Only concepts that can be resolved into ICD-9-CM codes (through synonymy or built-in mapping) are selected. For example, starting from C0239574 *Low grade pyrexia*, a mapping is found to ICD-9-CM code 780.6 *Fever*, which is contained in the concept C0015967, one of the ancestors of C0239574.

The last attempt to find a mapping involves not only hierarchical, but also associative relations. Instead of starting from the concept identified by MetaMap, ***associative mappings*** explore the concepts in associative relation to this concept. For example, the concept C1458136 *Renal stone substance* is mapped to ICD-9-CM code 592.0 *Calculus of kidney*.

Finally, when the identified ICD-9-CM code was not a leaf code (e.g., 786.5 *Chest pain*), we remapped it to one of the corresponding leaf codes in the training set where possible (e.g., 786.50 *Unspecified chest pain*).

Of the 2,331 UMLS concepts identified by MetaMap in the test set after freezing the method, 620 (27%) were mapped to ICD-9-CM. More specifically, 101 concepts were mapped to one of the 45 target ICD-9-CM codes present in the training set. Of the 101 concepts, 40 were mapped through synonymy, 11 through built-in mappings, 40 through hierarchical mapping and 10 through associative mapping.

After the main MTI processing was completed, we applied a post-processing filter, restricting our results to the list of 94 valid combinations of ICD-9-CM codes provided in the training set (henceforth referred to as allowed combinations) and slightly emphasizing MetaMap results. Examples of the post-processing rules are:

- If MTI recommended 079.99 (Unspecified viral infection in conditions…) via either MetaMap or Related Citations, use 079.99, 493.90 (Asthma, unspecified type…), and 780.6 (Fever) for indexing. This is the only valid combination for this code based on the training corpus.
- Similarly, if MTI recommended "Enlargement of lymph nodes" (785.6) via the MetaMap path with a score greater then zero, use 785.6 and 786.2 (Cough) for indexing.

The best F-score (F = 0.83) for the MTI method was obtained on the training set using the negation-removed text. This was a slight improvement over using the original text (F = 0.82).

SVM. We utilized Yet Another Learning Environment[3] (YALE), an open source application developed for machine learning and data mining, to determine the data classification performance of support vector machine (SVM) learning on the

[3] See http://rapid-i.com.

training data. To prepare the Challenge data for analysis, we removed all stop words and created feature vectors for the free text extracted from the "CLINICAL_HISTORY" and "IMPRESSION" fields of the records. Since both the training and test Challenge data had a known finite number of individual ICD-9-CM labels (45) and distinct combinations of ICD-9-CM labels (94), the data was prepared both as feature vectors for 45 individual labels as well as a model with 94 combination labels. In addition, the feature vectors were created using both simple term frequency as well as inverse document frequency (IDF) weighting, where the weight is (1+log(term frequency))*(total documents/document frequency). There were thus a total of four feature vector datasets: 1) 45 individual ICD-9-CM labels and simple term frequency, 2) 45 ICD-9-CM labels and IDF weighting, 3) 94 ICD-9-CM combinations and simple term frequency, and 4) 94 ICD-9-CM combinations and IDF weighting.

The YALE tool encompasses a number of SVM learners and kernel types. For the classification problem at hand, we chose the C-SVM learner and the radial basis function (rbf) kernel. The C-SVM learner attempts to minimize the error function

$$\frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \quad ,$$

$$\gamma_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, \ldots, N$$

where w is the vector of coefficients, b is a constant, φ is the kernel function, x are the independent variables, and ξ_i are parameters for handling the inputs. $C > 0$ is the penalty parameter of the error function. The rbf kernel is defined as $K(x, x') = \exp(-\gamma |x - x'|^2)$, $\gamma > 0$ where γ is a kernel parameter that determines the rbf width. We ran cross-validation experiments using YALE on all training datasets and varying C (10, 100, 1000, 10000) and γ (0.01, 0.001, 0.0001, 0.00001) to determine the optimal C and γ combination. The cross-validation experiments generated classification models that were then applied to the complete training datasets to analyze the performance of the learner. The 94 ICD-9-CM combination and simple term frequency dataset with C = 10000 and γ = 0.01 had the best F-score at 0.86. The best F-score for the 94 ICD-9-CM combination and IDF weight dataset was 0.79, where C = 0.001 and γ = 10000.

Further preprocessing the training dataset by removing negated expressions was found to improve the best F-score from 0.86 to 0.87. The C = 10000 and γ = 0.01 combination was then applied to the test dataset, which was preprocessed to remove negation and stop words and transformed to a feature vector using 94 ICD-9-CM combinations and simple term weighting. The predicted ICD-9-CM classifications and confidence of the predictions for each clinical free text report were output and later combined with other methods to optimize the accuracy and precision of our ICD-9-CM classifications.

k-NN. The Challenge training set was used to build a k-NN classifier. The k-NN classification method works by identifying, within a labelled set, documents similar to the document being classified, and inferring a classification for it from the labels of the retrieved neighbors.

The free text in the training data set was processed to obtain a vector-space representation of the patient reports.

Several methods of obtaining this representation were tested: after stop words were removed, simple term frequency and inverse document frequency (IDF) weighting were applied alternatively. A higher weight was also given to words appearing in the history portion of the text (*vs.* impression). Eventually, the most efficient representation was obtained by using controlled vocabulary terms extracted from the free text with MetaMap.[4] Further processing on this representation of the training data showed that removing negated portions of the free text improved the results, raising the F-score from 0.76 to 0.79.

Other parameters were also assessed on the training data, such as the number of neighbors to use (2 was found to be the best *vs.* 5, 10 or 15) and the restriction of the ICD-9-CM predictions to the set of 94 allowed combinations. When the prediction for a given document was not within the set of allowed 94 combinations, an allowed subset of the ICD-9-CM codes predicted was selected based on the individual scores obtained for each ICD-9-CM code.

The best F-score (F = 0.79) obtained on the training set used the MetaMap-based representa-

[4] Note that this use of MetaMap is independent of its inclusion as a component of MTI.

tion with simple frequency counts on the text with negated expressions removed. ICD-9-CM predictions were obtained from the nearest neighbors and restricted to one of the 94 allowed combinations.

Pattern Matching. We developed a pattern-matching classifier as a baseline for our more sophisticated classification methods. A list of all UMLS string representations for each of 45 codes (including synonyms from source vocabularies other than ICD-9-CM) was created as described in the MTI section above. The strings were then converted to lower case, punctuation was removed, and strings containing terms unlikely to be found in a clinical report were pruned. For example, *Abdomen NOS pain* and *Abdominal pain (finding)* were reduced to *abdominal pain*. For the same reasons, some of the strings were relaxed into patterns. For example, it is unlikely to see *PAIN CHEST* in a chart, but very likely to find *pain in chest*. The string, therefore, was relaxed to the following pattern: *pain.*chest*. The text of the clinical history and the impression fields of the radiology reports with negated expressions removed (see Section 2.2) was broken up into sentences. Each sentence was then searched for all available patterns. A corresponding code was assigned to the document for each matched pattern. This pattern matching achieved F-score = 0.79 on the training set. To reduce the number of codes assigned to a document, a check for allowed combinations was added as a post-processing step. The combination of assigned codes was looked up in the table of allowed codes. If not present, the codes were reduced to the combination of assigned codes most frequently occurring in the training set. This brought the F-score up to 0.84 on the training data. As the performance of this classifier was comparable to other methods, we decided to include these results when combining the predictions of the other classifiers.

2.3 Fusion of Basic Methods: Stacking

Experience with ad hoc retrieval tasks in the TREC Genomics Track has shown that combining predictions of several classifiers either significantly improves classification results, or at least provides more consistent and stable results when the training data set is small (Aronson et al., 2005). We therefore experimented with stacking (Ting and Witten, 1997), using a simple majority vote and a union of all assigned codes as baselines. The predictions of base classifiers described in the previous section were combined using our re-implementation of the stacked generalization proposed by Ting and Witten.

3 Results

Table 1 shows the results obtained for the training set. The best stacking results were obtained using predictions of all four base classifiers on the text with deleted negated expressions and with checking for allowed combinations. We retained all final predictions with probability of being a valid code greater than 0.3. Checking for the allowed combinations for the ensemble classifiers degraded the F-score significantly.

Classifier	F-score
MTI	0.83
SVM	0.87 (x-validation)
k-NN	0.79 (x-validation)
Pattern Matching	0.84
Majority	0.82
Stacking	**0.89**

Table 1: Training results for each classifier, the majority and stacking

Since stacking produced the best F-score on the training corpus and is known to be more robust than the individual classifiers, the corresponding results for the test corpus were submitted to the Challenge submission website. The stacking results for the test corpus achieved an F-score of 0.85 and a secondary, cost-sensitive accuracy score of 0.83. For comparison purposes, 44 Challenge submissions had a mean F-score of 0.77 with a maximum of 0.89. Our F-score of 0.85 falls between the 70th and 75th percentiles.

4 Discussion

It is significant that it was fairly straightforward to port various methods developed for ad hoc MEDLINE citation retrieval, indexing and classification to the assignment of codes to clinical text. The modifications to MTI consisted of replacing Restrict to MeSH with Restrict to ICD-9-CM, training the Related Citations method on clinical text and replacing MTI's normal post-processing with a much simpler version. Preprocessing the text using

NegEx to remove negated expressions was a further modification of the overall approach.

It is noteworthy that a simple pattern-matching method performed as well as much more sophisticated methods in the effort to fuse results from several methods into a final outcome. This unexpected success might be explained by the following limitations of the Challenge.

Possible limitations on the extensibility of the current research arise from two observations: (1) the Challenge cases were limited to two relatively narrow topics, cough/fever/pneumonia and urinary/kidney problems; and (2) the clinical text was almost error-free, a situation that would not be expected in the majority of clinical text. It is possible that these conditions contributed to the success of the pattern-matching method but also caused anomalous behavior, such as the fact that simple frequency counts provided a better representation than IDF for the SVM and k-NN methods.

Finally, as a result of low confidence in the ICD-9-CM code assignment, no codes were assigned to 29 records in the test set. It is worthwhile to explore the causes for such null assignments. One of the reasons for low confidence could be the aggressive pruning of the text by the negation algorithm. For example, after removal of negated text in the sample report given in section 2.1, the only remaining text is "13-year 2-month - old female" from the clinical history field; this provided no evidence for code assignment. Secondly, in some cases the original text was not sufficient for confident code assignment. For example, for the document with clinical history "Bilateral grade 3." and impression "Interval growth of normal appearing Kidneys", no code was assigned by the SVM, k-NN, or pattern-matching classifiers. Code 593.70 corresponding to the UMLS concept *Vesicoureteral reflux with reflux nephropathy, unspecified or without reflux nephropathy* was assigned by MTI with a very low confidence, which was not sufficient for the final assignment of the code. The third reason for assigning no code to a document was the wide range of assignments provided by the base classifiers. For example, for the following document: "CLINICAL_HISTORY: 3-year - old male with history of left ureteropelvic and ureterovesical obstruction. Status post left pyeloplasty and left ureteral reimplantation. IMPRESSION: 1. Stable appearance and degree of hydronephrosis involving the left kidney. Stable urothelial thickening. 2. Interval growth of kidneys, left greater than right. 3. Normal appearance of the right kidney with interval resolution of right urothelial thickening." MTI assigned codes 593.89 *Other specified disorders of kidney and ureter* and 591 *Hydronephrosis*. Codes 593.70 *Vesicoureteral reflux with reflux nephropathy, unspecified or without reflux nephropathy* and 753.3 *Double kidney with double pelvis* were assigned by the k-NN classifier. Pattern matching resulted in assignment of code 591 with fairly low confidence. No code was assigned to this document by the SVM classifier. Despite failing to assign codes to these 29 records, the conservative approach (using threshold) resulted in better performance, achieving F-score 0.85 compared to F-score 0.80 when all 1,634 codes assigned by the base classifiers were used.

5 Conclusion

We are left with two conclusions. First, this research confirms that combining several complementary methods for accomplishing tasks, ranging from ad hoc retrieval to categorization, produces results that are better and more stable than the results for the contributing methods. Furthermore, we have shown that the basic methods employing domain knowledge and advanced statistical algorithms are applicable to clinical text without significant modification. Second, although there are some limitations of the current Challenge test collection of clinical text, we appreciate the efforts of the Challenge organizers in the creation of a test collection of clinical text. This collection provides a unique opportunity to apply existing methods to a new and important domain.

Acknowledgements

This work was supported in part by the Intramural Research Program of the NIH, National Library of Medicine and by appointments of Aurélie Névéol and Vivian Lee to the NLM Research Participation Program sponsored by the National Library of Medicine and administered by the Oak Ridge Institute for Science and Education.

The authors gratefully acknowledge the many essential contributions to MTI, especially W. John Wilbur for the PubMed Related Citations indexing method, and Natalie Xie for adapting TexTool (an interface to Related Citations) for this paper.

References

Aronson AR, Demner-Fushman D, Humphrey SM, Lin J, Liu H, Ruch P, Ruiz ME, Smith LH, Tanabe LK, Wilbur WJ. Fusion of knowledge-intensive and statistical approaches for retrieving and annotating textual genomics documents. Proc TREC 2005, 36-45.

Aronson AR, Mork JG, Gay CW, Humphrey SM and Rogers WJ. The NLM Indexing Initiative's Medical Text Indexer. Medinfo. 2004: 268-72.

Bodenreider O, Nelson SJ, Hole WT and Chang HF. Beyond synonymy: exploiting the UMLS semantics in mapping vocabularies. Proc AMIA Symp 1998: 815-9.

Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan B. Evaluation of negation phrases in narrative clinical reports. Proc AMIA Symp. 2001a:105-9.

Chapman WW, Bridewell W, Hanbury P, Cooper GF and Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform. 2001b;34:301-10.

Demner-Fushman D, Humphrey SM, Ide NC, Loane RF, Ruch P, Ruiz ME, Smith LH, Tanabe LK, Wilbur WJ and Aronson AR. Finding relevant passages in scientific articles: fusion of automatic approaches vs. an interactive team effort. Proc TREC 2006, 569-76.

Fung KW and Bodenreider O. Utilizing the UMLS for semantic mapping between terminologies. AMIA Annu Symp Proc 2005: 266-70.

Gay CW, Kayaalp M and Aronson AR. Semi-automatic indexing of full text biomedical articles. AMIA Annu Symp Proc. 2005:271-5.

Goldin I and Chapman WW. Learning to detect negation with 'not' in medical texts. Proc Workshop on Text Analysis and Search for Bioinformatics, ACM SIGIR, 2003.

Hunter L and Cohen KB. Biomedical language processing: what's beyond PubMed? Mol Cell. 2006 Mar 3;21(5):589-94.

Tanabe L and Wilbur WJ. (2002) Tagging gene and protein names in biomedical text. *Bioinformatics*, Aug 2002; 18: 1124 –32.

Ting WK and Witten I. 1997. Stacking bagged and dagged models. 367-375. Proc. of ICML'97. Morgan Kaufmann, San Francisco, CA.

Appendix

A sample of the input to NegEx for dictionary generation:

```
C0002390 pneumonitis, allergic interstitial
C0002390 allergic interstitial pneumonitis, nos
C0002390 extrinsic allergic bronchiolo alveolitis
C0002390 extrinsic allergic bronchiolo alveolitis, nos
C0002390 hypersensitivity pneumonia
C0002390 hypersensitivity pneumonia, nos
C0002390 eaa  extrinsic allergic alveolitis
C0002390 allergic extrinsic alveolitis nos (disorder)
C0002390 extrinsic allergic alveolitis (disorder)
C0002390 hypersensitivity pneumonitis nos (disorder)
```

A sample of the dictionary generated by NegEx for later use in detecting negated expressions:

```
C0002098 hypersensitivity granuloma (morphologic abnormality
C0151726 hypersensitivity injection site
C0020517 hypersensitivity nos
C0429891 hypersensitivity observations
C0002390 hypersensitivity pneumonia
C0002390 hypersensitivity pneumonia, nos
C0002390 hypersensitivity pneumonitides
C0005592 hypersensitivity pneumonitides, avian
C0002390 hypersensitivity pneumonitis
C0182792 hypersensitivity pneumonitis antibody determination re-
agents
```

Automatically Restructuring Practice Guidelines using the GEM DTD

Amanda Bouffier **Thierry Poibeau**
Laboratoire d'Informatique de Paris-Nord
Université Paris 13 and CNRS UMR 7030
99, av. J.-B. Clément – F-93430 Villetaneuse
firstname.lastname@lipn.univ-paris13.fr

Abstract

This paper describes a system capable of semi-automatically filling an XML template from free texts in the clinical domain (practice guidelines). The XML template includes semantic information not explicitly encoded in the text (pairs of conditions and actions/recommendations). Therefore, there is a need to compute the exact scope of conditions over text sequences expressing the required actions. We present a system developed for this task. We show that it yields good performance when applied to the analysis of French practice guidelines.

1 Introduction

During the past years, clinical practices have considerably evolved towards standardization and effectiveness. A major improvement is the development of practice guidelines (Brownson *et al.*, 2003). However, even if widely distributed to hospitals, doctors and other medical staff, clinical practice guidelines are not routinely fully exploited[1]. There is now a general tendency to transfer these guidelines to electronic devices (via an appropriate XML format). This transfer is justified by the assumption that electronic documents are easier to browse than paper documents.

However, migrating a collection of texts to XML requires a lot of re-engineering. More precisely, it means analyzing the full set of textual documents so that they can fit with strict templates, as required either by XML schemas or DTD (document type definition). Unfortunately, most of the time, the semantic blocks of information required by the XML model are not explicitly marked in the original text. These blocks of information correspond to discourse structures.

This problem has thus renewed the interest for the recognition and management of discourse structures, especially for technical domains. In this study, we show how technical documents belonging to a certain domain (namely, clinical practice guidelines) can be semi-automatically structured using NLP techniques. Practice guidelines describe best practices with the aim of guiding decisions and criteria in specific areas of healthcare, as defined by an authoritative examination of current evidence (evidence-based medicine, see `Wikipedia` or Brownson *et al.*, 2003).

The Guideline Elements Model (GEM) is an XML-based guideline document model that can store and organize the heterogeneous information contained in practice guidelines (Schiffman, 2000). It is intended to facilitate translation of natural language guideline documents into a format that can be processed by computers. The main element of GEM, `knowledge component`, contains the most useful information, especially sequences of conditions and recommendations. Our aim is thus to format these documents which have been written manually without any precise model, according to the GEM DTD (see annex A).

The organization of the paper is as follows: first, we present the task and some previous approaches (section 2). We then describe the different processing steps (section 3) and the implementation (section 4). We finish with the presentation of some results (section 5), before the conclusion (section 6).

[1] See (Kolata, 2004). This newspaper article is a good example of the huge social impact of this research area.

BioNLP 2007: Biological, translational, and clinical language processing, pages 113–120,
Prague, June 2007.

2 Document Restructuring: the Case of Practice Guidelines

As we have previously seen, practice guidelines are not routinely fully exploited. One reason is that they are not easily accessible to doctors during consultation. Moreover, it can be difficult for the doctor to find relevant pieces of information from these guides, even if they are not very long. To overcome these problems, national health agencies try to promote the electronic distribution of these guidelines (so that a doctor could check recommendations directly from his computer).

2.1 Previous Work

Several attempts have already been made to improve the use of practice guidelines: for example knowledge-based diagnostic aids can be derived from them (*e.g*. Séroussi *et al*., 2001).

GEM is an intermediate document model, between pure text (paper practice guidelines) and knowledge-based models like GLIF (Peleg *et al.*, 2000) or EON (Tu and Musen, 2001). GEM is thus an elegant solution, independent from any theory or formalisms, but compliant with other frameworks.

GEM Cutter (`http://gem.med.yale.edu/`) is a tool aimed at aiding experts to fill the GEM DTD from texts. However, this software is only an interface allowing the end-user to perform the task through a time-consuming cut-and-paste process. The overall process described in Shiffman *et al.* (2004) is also largely manual, even if it is an attempt to automate and regularize the translation process.

The main problem in the automation of the translation process is to identify that a list of recommendations expressed over several sentences is under the scope of a specific condition (conditions may refer to a specific pathology, a specific kind of patients, temporal restrictions, etc.). However, previous approaches have been based on the analysis of isolated sentences. They do not compute the exact scope of conditional sequences (Georg and Jaulent, 2005): this part of the work still has to be done by hand.

Our automatic approach relies on work done in the field of discourse processing. As we have seen in the introduction, the most important sequences of text to be tagged correspond to discourse structures (conditions, actions …). Although most researchers agree that a better understanding of text structure and text coherence could help extract knowledge, descriptive frameworks like the one developed by Halliday and Hasan[2] are poorly formalized and difficult to apply in practice.

Some recent works have proposed more operational descriptions of discourse structures (Péry-Woodley, 1998). Several authors (Halliday and Matthiessen, 2004; Charolles, 2005) have investigated the use of non-lexical cues for discourse processing (*e.g* temporal adverbials like "*in 1999"*). These adverbials introduce situation frames in a narrative discourse, that is to say a 'period' in the text which is dependent from the adverbial.

We show in this study that condition sequences play the same role in practice guidelines: their scope may run over several dependent clauses (more precisely, over a set of several recommendations). Our plan is to automatically recognize these using surface cues and processing rules.

2.2 Our Approach

Our aim is to semi-automatically fill a GEM template from existing guidelines: the algorithm is fully automatic but the result needs to be validated by experts to yield adequate accuracy. Our system tries to compute the exact scope of conditional sequences. In this paper we apply it to the analysis of several French practice guidelines.

The main aim of the approach is to go from a textual document to a GEM based document, as shown on Figure 1 (see also annex A). We focus on conditions (including temporal restrictions) and recommendations since these elements are of paramount importance for the task. They are especially difficult to deal with since they require to accurately compute the scope of conditions.

The example on figure 1 is complex since it contains several levels of overlapping conditions. We observe a first opposition (*Chez le sujet non immunodéprimé* / *chez le sujet immunodéprimé… Concerning the non-immuno-depressed patient* / *Concerning the immuno-depressed patient…*) but a second condition interferes in the scope of this first level (*En cas d'aspect normal de la muqueuse iléale… If the ileal mucus seems normal…*). The task involves recognizing these various levels of conditions in the text and explicitly representing them through the GEM DTD.

[2] See "the text-forming component in the linguistic system" in Halliday and Hasan (1976:23).

Chez le sujet non immunodéprimé, **en cas d'aspect macroscopique normal de la muqueuse colique**, des biopsies coliques nombreuses et étagées sont recommandées (…). Les biopsies isolées sont insuffisantes (…).
L'exploration de l'iléon terminal est également recommandée (grade C). **En cas d'aspect normal de la muqueuse iléale (…)**, la réalisation de biospsies n'est pas systématique (accord professionnel).

Chez le sujet immunodéprimé, il est nécessaire de réaliser des biopsies systématiques (…)

<recommandation>
<decision.variable>Chez le sujet non immunodéprimé </decsion.variable>
<decision.variable>en cas d'aspect macroscopique normal de la muqueuse colique </decison.variable>
<action> des biopsies coliques nombreuses et étagées sont recommandées (…) </action>
<action>Les biopsies isolées sont insuffisantes(..) </action>
<action>L'exploration de l'iléon terminal est également recommandée</action>
</recommandation>

<recommandation>
<decision.variable>Chez le sujet non immunodéprimé </decsion.variable>
<decision.variable>en cas d'aspect macroscopique normal de la muqueuse colique </decison.variable>
<decision.variable>En cas d'aspect normal de la muqueuse iléale</decision.variable>
<action>la réalisation de biospsies n'est pas systématique</action>
</recommandation>

<recommandation
<decision.variable>Chez le sujet immunodéprimé</decision.variable>
<action> il est nécessaire de réaliser des biopsies systématiques(…)</action>
</recommandation>

Figure 1. From the text to GEM

What is obtained in the end is a tree where the leaves are recommendations and the branching nodes correspond to the constraints on conditions.

2.3 Data

We analyzed 18 French practice guidelines published by French national health agency (ANAES, *Agence Nationale d'Accréditation et d'Evaluation en Santé* and AFSSAPS, *Agence Francaise de Sécurité Sanitaire des Produits de Santé)* between 2000 and 2005. These practice guidelines focus on different pathologies (e.g. diabetes, high blood pressure, asthma etc.) as well as with clinical examination processes (e.g. digestive endoscopy). The data are thus homogeneous, and is about 250 pages long (150,000+ words). Most of these practice guidelines are publicly available at: `http://www.anaes.fr` or `http://affsaps.sante.fr`. Similar documents have been published in English and other languages; the GEM DTD is language independent.

3 Processing Steps

Segmenting a guideline to fill an XML template is a complex process involving several steps. We describe here in detail the most important steps (mainly the way the scope of conditional sequences is computed), and will only give a brief overview of the pre-processing stages.

3.1 Overview

A manual study of several French practice guidelines revealed a number of trends in the data. We observed that there is a default structure in these guidelines that may help segmenting the text accurately. This default segmentation corresponds to a highly conventionalized writing style used in the document (a *norm*). For example, the location of conditions is especially important: if a condition occurs at the opening of a sequence (a paragraph, a section…), its scope is by default the entire following text sequence. If the condition is included in the sequence (inside a sentence), its default scope is restricted to the current sentence (Charolles, 2005 for similar observations on different text types).

This default segmentation can be revised if some linguistic cues suggest another more accurate segmentation (*violation of the norm*). We make use of Halliday's theory of text cohesion (Halliday and Hasan, 1976). According to this theory, some "cohesion cues" suggest extending the default segmentation while some others suggest limiting the scope of the conditional sequence (see section 3.4).

3.2 Pre-processing (Cue Identification)

The pre-processing stage concerns the analysis of relevant linguistic cues. These cues vary in nature: they can be based either on the material structure or the content of texts. We chose to mainly focus on task-independent knowledge so that the method is portable, as far as possible (we took inspiration from Halliday and Matthiessen's introduction to functional grammar, 2004). Some of these cues

(especially connectors and lexical cues) can be automatically captured by machine learning methods.

Material structure cues. These features include the recognition of titles, section, enumerations and paragraphs.

Morpho-syntactic cues. Recommendations are not expressed in the same way as conditions from a morpho-syntactic point of view. We take the following features into account:

- *Part of speech tags*. For example *recommandé* should be a verb and not a noun, even if the form is ambiguous in French;
- *Tense and mood of the verb*. Present and future tenses are relevant, as well as imperative and conditional moods. Imperative and future always have an injunctive value in the texts. Injunctive verbs (see *lexical* cues) lose their injunctive property when used in a past tense.

Anaphoric cues. A basic and local analysis of anaphoric elements is performed. We especially focused on expressions such as *dans ce cas*, *dans les N cas précédents* (*in this case, in the n preceding cases…*) which are very frequent in clinical documents. The recognition of such expressions is based on a limited set of possible nouns that occurred in context, together with specific constraints (use of demonstrative pronouns, etc).

Conjunctive cues (discourse connectors). Conditions are mainly expressed through conjunctive cues. The following forms are especially interesting: forms prototypically expressing conditions (*si, en cas de, dans le cas où… if, in case of…*); Forms expressing the locations of some elements (*chez, en présence de... in presence of...*); Forms expressing a temporal frame (*lorsque, au moment où, avant de… when, before…*)

Lexical cues. Recommendations are mainly expressed through lexical cues. We have observed forms prototypically expressing recommendations (*recommander, prescrire, ... recommend, prescribe*), obligations (*devoir, ... shall*) or options (*pouvoir, ... can*). Most of these forms are highly ambiguous but can be automatically acquired from an annotated corpus. Some expressions from the medical domains can be automatically extracted using a terminology extractor (we use Yatea, see section 4, "Implementation").

3.3 Basic Segmentation

A *basic segment* corresponds to a text sequence expressing either a condition or a recommendation. It is most of the time a sentence, or a proposition inside a sentence.

Some of the features described in the previous section may be highly ambiguous. For this reason basic segmentation is rarely done according to a single feature, but most of the time according to a bundle of features acquired from a representative corpus. For example, if a text sequence contains an *injunctive* verb with an infinitive form at the beginning of a sentence, the whole sequence is typed as `action`. The relevant sets of co-occurring features are automatically derived from a set of annotated practice guidelines, using the chi-square test to calculate the dissimilarity of distributions.

After this step, the text is segmented into typed basic sequences expressing either a recommendation or a condition (the rest of the text is left untagged).

3.4 Computing Frames and Scopes

As for quantifiers, a conditional element may have a *scope* (a *frame*) that extends over several basic segments. It has been shown by several authors (Halliday and Matthiessen, 2004; Charolles, 2005) working on different types of texts that conditions detached from the sentence have most of the time a scope beyond the current sentence whereas conditions included in a sentence (but not in the beginning of a sentence) have a scope which is limited to the current sentence. Accordingly we propose a two-step strategy: 1) the default segmentation is done, and 2) a revision process is used to correct the main errors caused by the default segmentation (corresponding to the norm).

Default Segmentation

We propose a strategy which makes use of the notion of default. By default:

1. Scope of a heading goes up to the next heading;
2. Scope of an enumeration's header covers all the items of the enumeration ;
3. If a conditional sequence is detached (in the beginning of a paragraph or a sentence), its scope is the whole paragraph;
4. If the conditional sequence is included in a sentence, its scope is equal to the current sentence.

Cases 3 and 4 cover 50-80% of all the cases, depending on the practice guidelines used. However, this default segmentation is revised and modified when a linguistic cue is a continuation mark within the text or when the default segmentation seems to contradict some cohesion cue.

Revising the Default Segmentation

There are two cases which require revising the default segmentation: 1) when a cohesion mark indicates that the scope is larger than the default unit; 2) when a rupture mark indicates that the scope is smaller. We only have room for two examples, which, we hope, give a broad idea of this process.

1) Anaphoric relations are strong cues of text coherence: they usually indicate the continuation of a frame after the end of its default boundaries.

L'indication d'une insulinothérapie est recommandée **lorsque l'HbA1c est > 8%, sur deux contrôles successifs sous l'association de sulfamides/metformine à posologie optimale.** Elle est laissée à l'appréciation par le clinicien du rapport bénéfices/inconvénients de l'insulinothérapie **lorsque l'HbA1c est comprise entre 6,6% et 8% sous la même association.** Dans les deux cas, la diététique aura au préalable été réévaluée et un facteur intercurrent de décompensation aura été recherchée (accord professionnel).

Stratégie de prise en charge du patient diabétique de type 2 à l'exclusion de la prise en charge des complications (2000)

Figure 2. The last sentence introduced by *dans les deux cas* is under the scope of the conditions introduced by *lorsque*[3].

In Figure 2, the expression *dans les deux cas* (*in the two cases*…) is an anaphoric mark referring to the two previous utterances. The scope of the conditional segment introduced by *lorsque* (that would normally be limited to the sentence it appears in) is thus extended accordingly.

2) Other discourse cues are strong indicators that a frame must be closed before its default boundaries. These cues may indicate some contrastive, corrective or adversative information (*cependant, en revanche… however*). Justifications cues (*en effet, en fait … in effect*) also pertain to this class since a justification is not part of the `action` element of the GEM DTD.

Figure 3 is a typical example. The linguistic cue *en effet* (*in effect*) closes the frame introduced by

Chez les patients ayant initialement une concentration très élevée de LDL-cholestérol, et notamment chez les patients à haut risque dont la cible thérapeutique est basse (<1g/l), le prescripteur doit garder à l'esprit que la prescription de statine à fortes doses ou en association nécessite une prise en compte au cas par cas du rapport bénéfice/risque et ne doit jamais être systématique. En effet, les fortes doses de statines et les bithérapies n'ont pas fait l'objet à ce jour d'une évaluation suffisante dans ces situations.

(Prise en charge thérapeutique du patient dyslipidémique, 2005, p4)

Figure 3. The last sentence contains a justification cue (*en* effet) which limits the scope of the condition in the preceding sentence.

Chez les patients ayant initialement...(<1g/l) since this sequence should fill the `explanation` element of the GEM DTD and is not an `action` element.

4 Implementation

Accurate discourse processing requires a lot of information ranging from lexical cues to complex co-occurrence of different features. We chose to implement these in a classic blackboard architecture (Englemore and Morgan, 1988). The advantages of this architecture for our problem are easy to grasp: each linguistic phenomenon can be treated as an independent agent; inference rules can also be coded as specific agents, and a facilitator controls the overall process.

Basic linguistic information is collected by a set of modules called "linguistic experts". Each module is specialized in a specific phenomenon (text structure recognition, part-of-speech tagging, term spotting, *etc.*). The text structure and text formatting elements are recognized using Perl scripts. Linguistic elements are encoded in local grammars, mainly implemented as finite-state transducers (Unitex[4]). Other linguistic features are obtained using publicly available software packages, *e.g.* a part-of-speech tagger (Tree Tagger[5]) and a term extractor (Yatea[6]), *etc*. Each linguist expert is encapsulated and produces annotations that are stored in the database of facts, expressed in Prolog (we thus avoid the problem of overlapping XML tags, which are frequent at this stage). These annotations are indexed according to the textual clause they appear in, but linear ordering of the text is not cru-

[3] In figures 2 and 3, bold and grey background are used only for sake of clarity; actual documents are made of text without any formatting.

[4] http://www-igm.univ-mlv.fr/~unitex/

[5] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

[6] http://www-lipn.univ-paris13.fr/~hamon/YaTeA

cial for further processing steps since the system mainly looks for co-occurrences of different cues. The resulting set of annotations constitutes the "working memory" of the system.

Another set of experts then combine the initial disseminated knowledge to recognize basic segments (section 3.3) and to compute scopes and frames (section 3.4). These experts form the "inference engine" which analyzes information stored in the working memory and adds new knowledge to the database. Even when linear order is irrelevant for the inference process new information is indexed with textual clauses, to enable the system to produce the original text along with annotation.

A facilitator helps to determine which expert has the most information needed to solve the problem. It is the facilitator that controls, for example, the application of default rules and the revision of the default segmentation. It controls the chalk, mediating among experts competing to write on the blackboard. Finally, an XML output is produced for the document, corresponding to a candidate GEM version of the document (no XML tags overlap in the output since we produce an instance of the GEM DTD; all potential remaining conflicts must have been solved by the supervisor). To achieve optimal accuracy this output is validated and possibly modified by domain experts.

5 Evaluation

The study is based on a corpus of 18 practice guidelines in French (several hundreds of frames), with the aid of domain experts. We evaluated the approach on a subset of the corpus that has not been used for training.

5.1 Evaluation Criteria

In our evaluation, a sequence is considered correct if the semantics of the sequence is preserved. For example *Chez l'obèse non diabétique (accord professionnel)* (*In the case of an obese person without any diabetes (professional approval)*), recognition is correct even if *professional approval* is not *stricto sensu* part of the condition. On the other hand, *Chez l'obèse* (*In the case of an obese person*) is incorrect. The same criteria are applied for recommendations.

We evaluate the scope of condition sequences by measuring whether each recommendation is linked with the appropriate condition sequence or not.

5.2 Manual Annotation and Inter-annotator Agreement

The data is evaluated against practice guidelines manually annotated by two annotators: a domain expert (a doctor) and a linguist. In order to evaluate inter-annotator agreement, conditions and actions are first extracted from the text. The task of the human annotators is then to (manually) build a tree, where each action has to be linked with a condition. The output can be represented as a set of couples (*condition – actions*). In the end, we calculate accuracy by comparing the outputs of the two annotators (# of common couples).

Inter-annotator agreement is high (157 nodes out of 162, i.e. above .96 agreement). This degree of agreement is encouraging. It differs from previous experiments, usually done using more heterogeneous data, for example, narrative texts. Temporals (like "*in 1999"*) are known to open a frame but most of the time this frame has no clear boundary. Practice guidelines should lead to actions by the doctor and the scope of conditions needs to be clear in the text.

In our experiment, inter-annotator agreement is high, especially considering that we required an agreement between an expert and non-expert. We thus make the simplified assumption that the scope of conditions is expressed through linguistic cues which do not require, most of the time, domain-specific or expert knowledge. Yet the very few cases where the annotations were in disagreement were clearly due to a lack of domain knowledge by the non-expert.

5.3 Evaluation of the Automatic Recognition of Basic Sequences

The evaluation of basic segmentation gives the following results for the condition and the recommendation sequences. In the table, P is precision; R is recall; P&R is the harmonic mean of precision and recall (P&R = (2*P*R) / (P+R), corresponding to a F-measure with a β factor equal to 1).

Conditions:

	Without domain knowledge	With domain knowledge
P	1	1
R	.83	.86
P&R	.91	.92

Recommendations:

	Without domain knowledge	With domain knowledge
P	1	1
R	.94	.95
P&R	.97	.97

Results are high for both conditions and recommendations.

The benefit of domain knowledge is not evident from overall results. However, this information is useful for the tagging of titles corresponding to pathologies. For example, the title *Hypertension artérielle* (*high arterial blood pressure*) is equivalent to a condition introduced by *in case of*... It is thus important to recognize and tag it accurately, since further recommendations are under the scope of this condition. This cannot be done without domain-specific knowledge.

The number of titles differs significantly from one practice guideline to another. When the number is high, the impact on the performance can be strong. Also, when several recommendations are dependent on the same condition, the system may fail to recognize the whole set of recommendations.

Finally, we observed that not all conditions and recommendations have the same importance from a medical point of view – however, it is difficult to quantify this in the evaluation.

5.4 Evaluation of the Automatic Recognition of the Scope of Conditions

The scope of conditions is recognized with accuracy above .7 (we calculated this score using the same method as for inter-annotator agreement, see section 5.2).

This result is encouraging, especially considering the large number of parameters involved in discourse processing. In most of successful cases the scope of a condition is recognized by the default rule (default segmentation, see section 3.4). However, some important cases are solved due to the detection of cohesion or boundary cue (especially titles).

The system fails to recognize extended scopes (beyond the default boundary) when the cohesion marks correspond to lexical items which are related (synonyms, hyponyms or hypernyms) or to complex anaphora structures (nominal anaphora; hyponyms and hypernyms can be considered as a special case of nominal anaphora). Resolving these rarer complex cases would require "deep" domain knowledge which is difficult to implement using state-of-art techniques.

6 Conclusion

We have presented in this paper a system capable of performing automatic segmentation of clinical practice guidelines. Our aim was to automatically fill an XML DTD from textual input. The system is able to process complex discourse structures and to compute the scope of conditional segments spanning several propositions or sentences. We show that inter-annotator agreement is high for this task and that the system performs well compared to previous systems. Moreover, our system is the first one capable of resolving the scope of conditions over several recommendations.

As we have seen, discourse processing is difficult but fundamental for intelligent information access. We plan to apply our model to other languages and other kinds of texts in the future. The task requires at least adapting the linguistic components of our system (mainly the pre-processing stage). More generally, the portability of discourse-based systems across languages is a challenging area for the future.

References

R.C. Brownson, E.A. Baker, T.L. Leet, K.N. Gillespie. 2003. *Evidence-based public health*. Oxford University Press. Oxford, UK.

M. Charolles. 2005. "Framing adverbials and their role in discourse cohesion: from connexion to forward labeling". *Papers of the Symposium on the Exploration and Modelling of Meaning* (Sem'05), Biarritz. France.

R. Englemore and T. Morgan. 1988. *Blackboard Systems*. Addison-Wesley, USA.

G. Georg and M.-C. Jaulent. 2005. "An Environment for Document Engineering of Clinical Guidelines". *Proceedings of the American Medical Informatics Association*. Washington DC. USA. pp. 276–280.

M.A.K. Halliday and R. Hasan. 1976. *Cohesion in English*. Longman. Harlow, UK.

M.A.K. Halliday and C. Matthiessen. 2004. *Introduction to functional grammar* (3rd ed.). Arnold. London, UK.

G. Kolata. 2004. "Program Coaxes Hospitals to See Treatments Under Their Noses". *The New York Times*. December 25, 2004.

M. Peleg, A. Boxwala, O. Ogunyemi, Q. Zeng, S. Tu, R. Lacson, E. Bernstam, N. Ash, P. Mork, L. Ohno-Machado, E. Shortliffe and R. Greenes. 2000. "GLIF3: The Evolution of a Guideline Representation Format". In *Proceedings of the American Medical Informatics Association*. pp. 645–649.

M-P. Péry-Woodley. 1998. "Signalling in written text: a corpus-based approach". In M. Stede, L. Wanner & E. Hovy (Eds.), *Proceeding of te Coling '98 Workshop on Discourse Relations and Discourse Markers*, pp. 79–85

B. Séroussi, J. Bouaud, H. Dréau., H. Falcoff., C. Riou., M. Joubert., G. Simon, A. Venot. 2001. "ASTI : A Guideline-based drug-ordering system for primary care". In *Proceedings MedInfo*. pp. 528–532.

R.N. Shiffman, B.T. Karras, A. Agrawal, R. Chen, L. Marenco, S. Nath. 2000. "GEM: A proposal for a more comprehensive guideline document model using XML". *Journal of the American Medical Informatics Assoc*. n°7(5). pp. 488–498.

R.N. Shiffman, M. George, M.G. Essaihi and E. Thornquist. 2004. "Bridging the guideline implementation gap: a systematic, document-centered approach to guideline implementation". In *Journal of the American Medical Informatics Assoc*. n°11(5). pp. 418–426.

S. Tu and M. Musen. 2001. "Modeling data and knowledge in the EON Guideline Architecture". In *Medinfo*. n°10(1). pp. 280–284.

Annex A. Screenshots of the system

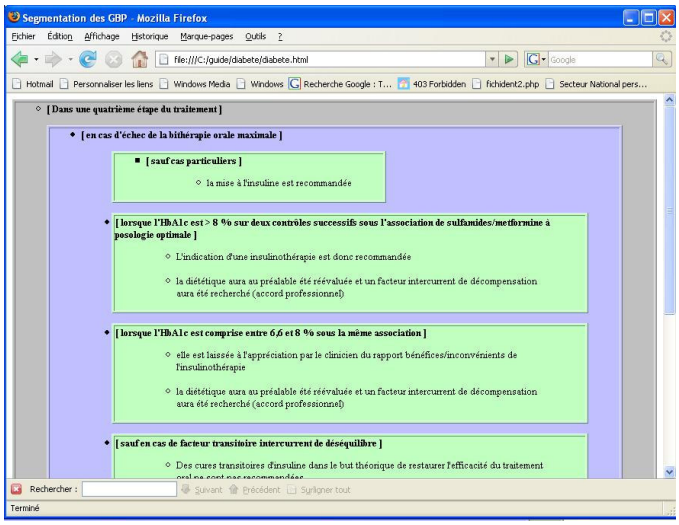
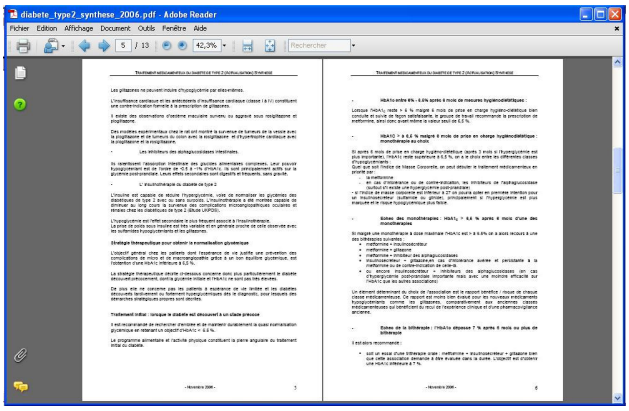
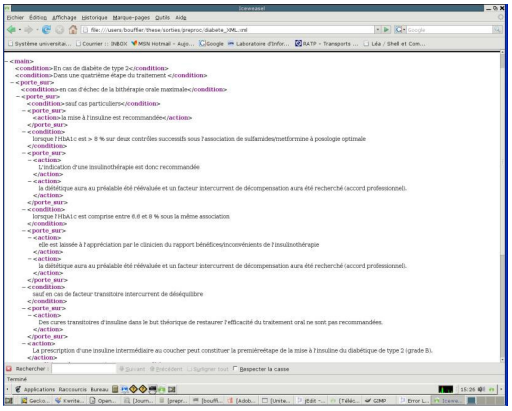


Figure A1. A practice guideline once analyzed by the system (*Traitement médicamenteux du diabète de type 2*, AFSSAPS-HAS, nov. 2006)





Figures A2 and A3. The original text, an the XML GEM template instanciated from the text

A Study of Structured Clinical Abstracts and the Semantic Classification of Sentences

Grace Y. Chung and Enrico Coiera
Centre for Health Informatics
University of New South Wales
Sydney NSW 2052 Australia
{graceyc, e.coiera}@unsw.edu.au

Abstract

This paper describes experiments in classifying sentences of medical abstracts into a number of semantic classes given by section headings in structured abstracts. Using conditional random fields, we obtain F-scores ranging from 0.72 to 0.97. By using a small set of sentences that appear under the PARTICPANTS heading, we demonstrate that it is possible to recognize sentences that describe population characteristics of a study. We present a detailed study of the structure of abstracts of randomized clinical trials, and examine how sentences labeled under PARTICIPANTS could be used to summarize the population group.

1 Introduction

Medical practitioners are increasingly applying evidence-based medicine (EBM) to support decision-making in patient treatments. The aim of EBM (Sackett, 1998) is to provide improved care leading to better outcomes through locating evidence for a clinical problem, evaluating the quality of the evidence, and then applying to a current problem at hand. However, the adoption of EBM is hampered by the overwhelming amount of information available, and insufficient time and skills on the clinician's part to locate and synthesize the best evidence in the scientific literature.

MEDLINE abstracts about randomized clinical trials (RCTs) play a critical role in providing the best evidence for the latest interventions for any given conditions. The MEDLINE database now has 16 million bibliographic entries, many of them include the abstract and more than 3 million of these were published in the last 5 years (Hunter, 2006).

To alleviate the information overload, some resources such as the Cochrane Collaboration (Cochrane, 2007), Evidence-Based Medicine (EBM, 2007), the ACP Journal Club (ACP, 2007) and BMJ Clinical Evidence (BMJCE, 2007), employ human experts to summarize knowledge within RCTs through extensive searches and critical assessments.

In (Sim, 2000), RCT information is entered into electronic knowledge bases or *"trial banks"*, easing the task for systematic reviewing and critical appraisal. This project requires manual entry of descriptions about the design and execution (subjects, recruitment, treatment assignment, follow-up), and hence, only small numbers of RCTs have been archived thus far.

The goal of our research is to use natural language processing to extract the most important pieces of information from RCTs for the purpose of automatic summarization, tailored towards the medical practitioner's clinical question at hand. Ultimately, it is our vision that data mined from full text articles of RCTs not only aid clinicians' assessments but researchers who are conducting meta-analyses.

In this paper, we examine the use of section headings that are frequently given in abstracts of medical journal articles. These section headings are topic-independent. Effectively they define the discourse structure for the abstract, and provide semantic labels to the sentences that fall under them.

BioNLP 2007: Biological, translational, and clinical language processing, pages 121–128,
Prague, June 2007.

Other researchers have recognized the potential utility of these heading (McKnight, 2003; Xu, 2006; Lin, 2006). It has also been recognized that scientific abstracts with such labels could be of importance to text summarization, information retrieval and question answering (Lee, 2006; Zweigenbaum, 2003). We share similar goals to previous research; the section headings of these structured medical abstracts can be used as training data for building labelers that can tag unstructured abstracts with discourse structure. But also, there is a large number of heading names. Sentences that occur under these heading names form a labeled training set which could be used to build a classifier that recognizes similar sentences. Ultimately, we would like to build finer-grained classifiers that exploit these semantic labels.

In our work, we seek to demonstrate that information about patient characteristics can now be extracted from structured and unstructured abstracts. We are motivated by the fact that patient characteristics is one of the fundamental factors most pertinent to evaluation of relevance to a clinical question. The total number of subjects in a trial reflects on the quality of the RCT, and additional factors such as age, gender and other co-existing conditions, will be crucial for assessing whether an RCT is relevant to the medical practitioner's immediate patient.

This paper is organized as follows. In Section 1 we will describe how the RCT abstracts were obtained, and we present a study of the discourse headings that occur in our document corpus. Section 3 will detail our sentence classification experiments. We first explore classification in which the abstracts are labeled under five subheadings, one of which describes the patients or population group. We also perform classification using a combined two-stage scheme, bootstrapping from partially labeled data. Finally in Section 4, we consider how well the PARTICIPANTS labeled sentences capture sentences containing the total number of participants in a trial. In Section 5, we will give a detailed analysis of the labeled sentences.

1. RESULTS	6. METHODS / RESULTS
2. METHODS	7. OBJECTIVE
3. CONCLUSION	8. PATIENTS / METHODS
4. BACKGROUND	9. PURPOSE
5. CONCLUSION	10. DESIGN

Table 1: The most common headings in RCT abstracts.

2 The Data

2.1 Corpus Creation

The current corpus is obtained by a MEDLINE search for RCTs. We did not constrain publications by their date. For the purpose of constraining the size of our corpus in these preliminary experiments, it was our intention to use RCTs pertaining to a fixed set of clinical conditions. Hence, we conducted a MEDLINE search for RCTs with the following keywords: asthma, diabetes, breast cancer, prostate cancer, erectile dysfunction, heart failure, cardiovascular, angina. The resultant corpus contains 7535 abstracts of which 4268 are structured.

2.2 Structure of Medical Abstracts

Structured abstracts were introduced in 1987 (Ad-Hoc, 2005) to help clinical readers to quickly select appropriate articles, and allow more precise information retrieval. However, currently, the majority of medical abstracts remain unstructured. Previous studies have concluded that while many scientific abstracts follow consistent patterns (e.g. Introduction, Problem, Method, Evaluation, Conclusion) many still contain missing sections or have differing structures (Orasan, 2001; Swales, 1990; Meyer, 1990). Journals vary widely in their requirements for abstract structures.

We have conducted a study of the structured abstracts in our corpus. Of 4268 structured abstracts, we have found a total of 238 unique section headings. The most common ones are shown in Table 1. To investigate the numbers of variations in the abstract structure, we first manually map headings that

Class	Example Heading Names
Aim	AIM, AIMS, AIM OF THE STUDY..
Setting	SETTING, SETTINGS, STUDY SETTING..
Participants	PARTICIPANTS, PATIENTS, SUBJECTS..
Setting/ Subjects	PARTICIPANTS AND SETTINGS, SETTING/PATIENTS..

Table 2: Examples of manual mappings for heading names into equivalence classes.

Structure of Abstracts	% of Corpus
BACKGROUND, METHOD, RESULT, CONCLUSION	16%
AIM, METHOD, RESULT, CONCLUSION	14%
AIM, PATIENT AND METHOD, RESULT, CONCLUSION	8.5%
BACKGROUND, AIM, METHOD, RESULT, CONCLUSION	7.6%
BACKGROUND, METHOD AND RESULTS, CONCLUSION	6.6%
AIM, PARTICIPANTS, DESIGN, MEASUREMENTS, RESULT, CONCLUSION	<1%
CONTEXT, DESIGN, SETTING, PARTICIPANTS, OUTCOME MEASURES, RESULT, CONCLUSION	<1%
AIM, DESIGN AND SETTING, PARTICIPANTS, INTERVENTION MEASUREMENTS AND MAIN RESULTS, CONCLUSION	<1%

Table 3: Examples of the patterns that occur in the section headings of structured RCT abstracts.

are essentially semantically equivalent to the same classes, resulting in 106 classes. Examples of these mappings are shown in Table 2. After the class mappings are applied, it turns out that there are still 400 different patterns in the combinations of section headings in these medical abstracts, with over 90% of these variations occurring less than 10 times. The most common section heading patterns are shown in Table 3. Some of the less common ones are also shown.

In studying the structure of these medical abstracts, we find that the variation in structural ordering is large, and many of the heading names are unique, chosen at the discretion of the paper author. Some of the most frequent heading names are also compound headings such as: METHODS/RESULTS, RESULTS/CONCLUSION, PATIENTS/RESULTS, SUBJECTS AND SETTINGS.

3 Sentence Classification Experiments

3.1 Extracting Participant Sentences

In this work, we seek to build a classifier using training data from the semantic labels already provided by structured abstracts. It is our intention ultimately to label both structured and unstructured abstracts with the semantic labels that are of interest for the purposes of information extraction and answering specific questions regarding the trial. In our approach, we identify in our structured abstracts the ones with section headings about patient characteristics. These are collapsed under one semantic class and used as training data for a classifier.

From our 4268 structured abstracts, all the heading names are examined and are re-mapped by hand to one of five heading names: AIM, METHOD, PARTICIPANTS, RESULTS, CONCLUSION. Most heading names can be mapped to these general headings but the subset containing compound headings such as METHOD/RESULT are discarded.

All the abstracts are segmented into sentences and tokenized via Metamap (Aronson, 2001). Some abstracts are discarded due to sentence segmentation errors. The remainder (3657 abstracts) forms the corpus that we will work with here. These abstracts are randomly divided into a training set and an initial test set, and for purposes of our experiments, they are further subdivided into abstracts with the PARTICIPANTS label and those without. The exact size of our data sets are given in Table 4.

Although abstracts in Train Set A are generally structured as (AIM, METHOD, RESULTS, CONCLUSION), they contain sentences pertaining to patient or population group largely in the METHOD section. In the following, we will explore three ways for labeling sentences in the abstract including labeling for sentences that describe the population group. The first employs a 5-class classifier, the second uses a two-stage approach and the third employs an approach which uses partially labeled data.

3.2 Using Labeled Data Only

Using only abstracts from Train Set B, all sentences are mapped into one of 5 classes: AIM, PARTICIPANTS, METHOD, RESULTS, CONCLUSION.

Data Set	Number of Abstracts	Number of Sentences
Total in Corpus	3657	45k
Total Train Set	3439	42k
Train Set A (no PARTICIPANTS)	2643	32k
Train Set B (w/ PARTICIPANTS)	796	10k
Test Set (w/ PARTICIPANTS)	62	878

Table 4: Sizes of data sets.

	Recall	Precision	F-score
CRF	Accuracy = 84.4%		
Aim	0.98	0.91	0.95
Method	0.52	0.73	0.61
Participants	0.79	0.73	0.76
Results	0.95	0.87	0.91
Conclusion	0.91	0.97	0.94
SVM	Accuracy = 80.2%		
Aim	0.87	0.91	0.90
Method	0.64	0.68	0.67
Participants	0.73	0.70	0.72
Results	0.89	0.84	0.86
Conclusion	0.80	0.88	0.83

Table 5: Classification of sentences in RCT abstracts into 5 semantic classes using CRFs and SVMs. The recall, precision and F-score are reported on our unseen test set.

The PARTICIPANTS class subsume all headings that include mention of population characteristics. These include compound headings such as: SETTING/POPULATION, PATIENTS/DESIGN. Sentences associated with these compound headings often include long sentences that describe the participant group as well as a second aspect of the study such as setting or design.

We build a 5-class classifier using linear-chain conditional random fields (CRFs).[1] CRFs (Sutton, 2006) are undirected graphical models that are discriminatively trained to maximize the conditional probability of a set of output variables given a set of input variables. We simply use bag-of-words as features because past studies (McKnight, 2003), using n-gram-based features did not improve accuracies.[2]

As a baseline comparison, we have performed classification using a Support Vector Machine (SVM) classifier (Burges, 1998; Witten, 2005), with a radial basis functions (RBF) kernel. To help model the sequential ordering, a normalized integer for the sentence number in the abstract is included as a feature.

Experimental results are shown in Table 5. CRFs clearly outperform SVMs in this classification task. This may in part be attributable to the explicit sequential modeling in the CRFs compared with SVMs. While our training set (796 abstracts in Train set B) is substantially smaller than that reported in previous studies (McKnight, 2003; Lin, 2006; Xu, 2006), the F-score for AIM, RESULTS, CONCLUSION are comparable to previous results. By far the largest sources of classification error are the confusions between METHOD and PARTICIPANTS class. In training we have included into the PARTICIPANTS class all sentences that come under compound headings, and therefore the PARTICIPANTS section can often encompass several sentences that contain detailed information regarding the intervention, and the type of study, as exemplified below.

> Doppler echocardiography was performed in 21 GH deficient patients after 4 months placebo and 4 months GH therapy, in a double blind cross-over study. In an open design study, 13 patients were reinvestigated following 16 months and 9 patients following 38 months of GH therapy. Twenty-one age and sex-matched normal control subjects were also investigated.

Nonetheless, information about the patient population is embedded within these sentences.

3.3 Using a Two-Stage Method

An alternative approach is to adopt a two-stage hierarchical strategy. First we build a classifier which performs a 4-way classification based on the labels AIM, METHOD, RESULTS, CONCLUSION, and a second stage binary classifier tags all the METHOD sentences into either METHOD or PARTICIPANTS. There are two distinct advantages to this approach. (1) In our 5-class classifier, it is clear that METHOD and PARTICIPANTS are confusable and a dedicated classifier to perform this subtask may be more effective. (2) The corpus of abstracts with only the 4 classes labeled is much larger (3439 abstracts), and hence the resultant classifier is likely to be trained more robustly. Our first stage classifier is a CRF tagger. It is trained on the combined training sets A and B, whereby all sentences in the structured abstracts are mapped to the 4-class labels. The second stage binary classifier is an SVM classifier. The SVM classifier has been augmented with additional features of the semantic labels tagged via Metamap tagger. It is trained on the subset of Train Set A (3499 sentences) that is labeled as either METHOD or PARTICIPANTS.

Classification results for the unseen test set are reported in Table 6. The 4-class classifier yields F-scores between 0.92 and 0.96. We report results for

[1] We used the SimpleTagger command line interface of the Mallet software package (McCallum, 2002).

[2] In other experiments, attempts to use stemming and removal of stop words also did not improve performance.

(1) 4-class	Accuracy = 92.7%		
	Recall	Precision	F-score
Aim	0.98	0.94	0.96
Method	0.89	0.95	0.92
Results	0.95	0.89	0.92
Conclusion	0.91	0.97	0.94
(2) 2-class	Accuracy = 80.1%		
Method	0.73	0.83	0.78
Participants	0.87	0.78	0.81
(3) 5-class	Accuracy = 86.0%		
Aim	0.96	0.92	0.96
Method	0.66	0.79	0.71
Participants	0.77	0.72	0.75
Results	0.94	0.89	0.92
Conclusion	0.91	0.97	0.94

Table 6: (1) Classification using CRFs into 4 major semantic classes with combined Train Set A and B as training data. (2) Binary SVM classification of a subset of test set sentences. (3) Classification into 5 classes as described in Section 3.3. All results (recall, precision and F-score) are reported on the unseen test set.

the binary SVM classifier on the subset of test set sentences (253 sentences) that are either METHOD or PARTICIPANTS in Table 6.

The two stage method here has yielded some gains in performance for each class except for PARTICIPANTS. The gains are likely to have been due to increased training data particularly for the classes, AIM, RESULTS and CONCLUSION.

3.4 Augmenting with Partially Labeled Data

We investigate a second method for leveraging the data available in Train Set A. We hypothesize that many sentences within the METHOD section of Train Set A do in fact describe patient information and could be used as training data. We propose a bootstrapping method whereby some of the sentences in Train Set A are tagged by a binary SVM classifier and used as training data in the 5-class CRF classifier. The following describes each step:

1. A binary SVM classifier is trained on the subset of sentences in Train Set B labeled with METHOD and PARTICIPANTS.
2. The trained SVM classifier is used to label all the sentences in Train Set A that are originally labeled with the METHOD class.
3. All the sentences in Train Set A are now labeled in terms of the 5 classes, and a score is available from the SVM output is associated with those sentences labeled as either METHOD or PARTICIPANTS. The abstracts that contain sentences scoring above a pre-determined threshold score are then pooled with sentences in Train Set B into a single training corpus. We tuned the threshold value by testing on a development set held out from Train Set B. As a result, 1217 sentences from Train Set A is combined with Train Set B.
4. The final training corpus is used to train a CRF tagger to label sentences into one of 5 classes.

	Recall	Precision	F-score
5-class	Accuracy = 87.6%		
Aim	0.99	0.95	0.97
Method	0.67	0.77	0.72
Participants	0.90	0.77	0.83
Results	0.91	0.92	0.92
Conclusion	0.90	0.97	0.93

Table 7: Classification into 5 classes as described in Section 3.4. All results (recall, precision and F-score) are reported on the unseen test set.

The results of classification on the unseen test set are reported in Table 7. Overall accuracy for classification improves to 87.6% primarily because there is a marked improvement is observed for the F-scores of the PARTICIPANTS class. Our best results here are comparable to those previously reported on similar tasks on the class, AIM, RESULTS and CONCLUSION (Xu, 2006; Lin, 2006). The F-score for METHOD is lower because introducing a PARTICIPANTS label has increased confusability.

4 Extraction of Number of Patients

We have demonstrated that for a structured abstract it is possible to predict sentences that are associated with population characteristics. However, our ultimate objective is to extract these kinds of sentences from unstructured abstracts, and even to extract more fine-grained information. In this section, we will examine whether labeling sentences into one of 5 classes can aid us in the extraction of the total number of patients from an RCT.

	Abstracts w/ Total Subjects	% tagged as PARTICIPANTS
Structured	46	87%
Unstructured	103	72%

Table 8: Extraction of the total number of subjects in a trial in a human annotated test set, as described in Section 4.2

4.1 Annotation

In a concurrent annotation effort to label RCT abstracts, human annotators manually tagged a separate test set of 204 abstracts with the total number of participants in each study. Of the 204 abstracts, 148 are unstructured and 56 are structured. None of these 204 abstracts are part of the training set, described in this paper.

4.2 Experiments

The abstracts from this annotated test set are processed by the classifier described in Section 3.4. For all the abstracts which mention the total number of participants in the RCT, we compute the frequency for which this is included in the sentences labeled as PARTICIPANTS. Results are depicted in Table 8.

Upon subsequent examination of the test set, it is found that only 82% (46/56) of the structured abstracts and 70% (103/148) of unstructured abstracts contain information about total number of participants in the trial. As seen in Table 8, in 87% of the 46 structured abstracts, and in 72% of the 103 unstructured abstracts, the total number of participants are mentioned in the labeled PARTICIPANTS sentences. The extraction of the total number of participants is significantly worse in unstructured abstracts which do not adhere to the strict discourse structures given by the headings of structured abstracts. In 13% (13/103) of the unstructured abstracts, the total number of participants appears in the first sentence, which is usually tagged as the AIM. It is evident that in the absence of structure, patient information can occur in any sentence in the abstract, or for that matter, it may appear only in the body of the paper. Our method of training first on structured abstracts may be a strong limitation to extraction of information from unstructured abstracts.

Even for the structured abstracts in the test set, 9% (4/46) of the set of abstracts containing population number actually mention the number in the AIM or RESULTS section, rather than the METHOD or PARTICIPANTS. Only 12 abstracts contain explicit headings referring to participants, where the total number of subjects in the trial is mentioned under the corresponding heading.

In this task, we only consider that total number of subjects enrolled in a study, and have yet to account for additional population numbers such as the drop out rate, the follow-up rate, or the number of subjects in each arm of a study. These are often reported in an abstract without mentioning the total number of patients to begin with. The classifier will tag sentences that describe these as PARTICIPANT sentences nonetheless.

5 Analysis and Discussion

We will further analyze the potential for using sentences tagged as PARTICIPANTS as summaries of population characteristics for a trial. Table 9 gives some examples of sentences tagged by the classifier.

Sentences that appear under PARTICIPANTS in structured abstracts are often concise descriptions of the population group with details about age, gender, and conditions, as seen in Example 1. Otherwise, they can also be extensive descriptions, providing selection criteria and some detail about method, as in Example 2.

Examples 3 and 4 show sentences from the test set of Section 4. Example 3 has been labeled as a PARTICIPANTS sentence by the classifier. It describes patient characteristics, giving the population number for each arm of the trial but does not reveal the total number of subjects. Example 3 appears under the heading METHODS AND RESULTS in the original abstract. Example 4 is from an unstructured abstract, where information about the intervention and population and study design are interleaved in the same sentences but tagged by the classifier as PARTICIPANTS. Many sentences tagged as PARTICIPANTS also do not give explicit information about population numbers but only provide descriptors for patient characteristics.

It is also plausible that our task has been made more challenging compared with previous reported studies because our corpus has not been filtered for publication date. Hence, the numbers of publica-

1. Male smokers aged 50–69 years who had angina pectoris in the Rose chest pain questionnaire at baseline (n = 1795). *PMID: 9659191*
2. The study included 809 patients under 70 years of age with stable angina pectoris. The mean age of the patients was 59 +/- 7 years and 31% were women. Exclusion criteria were myocardial infarction within the previous 3 years and contraindications to beta-blockers and calcium antagonists. The patients were followed between 6 and 75 months (median 3.4 years and a total of 2887 patient years). *PMID: 8682134*
3. Subjects with Canadian Cardiovascular Society (CCS) class 3/4 angina and reversible perfusion defects were randomized to SCS (34) or PMR (34). *PMID: 16554313*
4. Sixty healthy women, half of whom had been using OCs for at least the previous 6 months, participated in the study. Approximately two thirds were smokers and were randomized to be tested after either a 12 hr nicotine deprivation or administration of nicotine gum. One third were nonsmokers. *PMID: 11495215*

Table 9: Examples of sentences labeled under PARTICIPANTS class, forming summaries of the population characteristics of a trial. Examples 1 and 2 are typical sentences under the PARTICIPANTS heading in the train set. Examples 3 and 4 are from the annotated test set. See Section 5 for more detailed explanation.

tions and structural characteristics of our abstracts may be broader than previous reports which filter for abstracts to a narrow time frame (Xu, 2006).

6 Related Work

In recent years, there has been a growth in research in information extraction and NLP in the medical domain particularly in the RCT literature. This is due in part to the emergence of lexical and semantic resources such as the Unified Medical Language System (UMLS) (Lindberg, 1993), and software such as MetaMap (Aronson, 2001), which transforms text into UMLS concepts, and SemRep (Rindflesch, 2003), which identifies semantic propositions.

There are a number of previous attempts to perform text categorization on sentences in MEDLINE abstracts into generic discourse level section headings. They all share the goal of assigning structure to unstructured abstracts for the purpose of summarization or question answering. All previous attempts have mapped the given headings to four or five generic classes, and performed text categorization on large sets of RCTs without any disease or condition-specific filtering. Studies have shown that results deteriorate when classifying sentences in unstructured abstracts (McKnight, 2003; Lin, 2006). In (McKnight, 2003), McKnight and Srinivisan used an SVM for tagging sentences into 4 classes. Using a corpus of 7k abstracts, they obtain F-scores from 0.82 to 0.89. Later papers in (Xu, 2006; Lin, 2006) have found that Hidden Markov Models (HMMs) based approaches more effectively model the sequential ordering of sentences in abstracts. In (Xu, 2006), several machine learning methods, decision tree, maximum entropy and naive Bayes, are evaluated with an HMM-based algorithm. 3.8k abstracts from 2004 and 2005 were used as training data, and experiments yielded average precision of 0.94 and recall of 0.93.

One driving model for information extraction in RCTs is the PICO framework (Richardson, 1995). This is a task-based model for EBM formulated to assist EBM practitioners to articulate well-formed questions in order to find useful answers in clinical scenarios. PICO elements are Patient/Population, Intervention, Comparison and Outcome. This model has been adopted by researchers (Demner-Fushman, 2005; Niu, 2004) as a guideline for elements that can be automatically extracted from RCTs and patient records. However, doubts have been raised about the utility of PICO as a generic knowledge representation for computational approaches to answering clinical questions (Huang, 2006).

In experiments reported in (Demner-Fushman, 2005), the PICO framework was used as a basis for extracting population, problem, intervention and comparison for the purpose of evaluating relevance of an abstract to a particular clinical question. In this work, the population statements were located via a set of hand-written rules that were based on extracting an actual numeric value for the population.

7 Conclusions

In this study, we investigated the use of conditional random fields for classifying sentences in medical abstracts. Our results particularly in terms of F-scores for generic section headings such as AIM, RE-

SULTS and CONCLUSION were comparable to previous studies, even with smaller training sets. We investigated the use of text classification by leveraging the subset of abstracts with explicitly labeled PARTICIPANTS sentences combining the use of CRFs and SVMs, and exploiting partially labeled data.

One main objective here is to label sentences that describe population characteristics in structured and unstructured abstracts. We found that unstructured abstracts differ substantially from structured ones, and alternative approaches will be necessary for extracting information from unstructured abstracts. Furthermore, critical details that are needed by a physician when evaluating a study such as exclusion criteria, drop out rate, follow up rate, etc, may only be listed in the full text of the study. Future work will address extracting information beyond the abstract.

8 Acknowledgment

The authors would like to acknowledge the anonymous reviewers and the executive committee for their comments and suggestions, and Marianne Byrne, Brenda Anyango Omune and Wei Shin Yu for annotation of the abstracts. This project is funded by the Australian Research Council, grant number DP0666600.

References

ACP Journal Club. Available from: http://www.acpjp.org

Ad Hoc working group for Critical Appraisal of the Medical Literature 1987. A proposal for more informative abstracts of clinical articles. *Annals of Int. Medicine* 106:595–604.

A. R. Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. *Ann. Symp. of AMIA* pp 17–21.

Clinical Evidence. BMJ Publishing Group. Available from: http://www.clinicalevidence.com

C. Burges. 1998. A Tutorial on Support Vector Machines for Pattern Recognition *Journal Data Mining and Knowledge Discovery*, 2(2), June.

The Cochrane Collaboration. Available from: http://www.cochrane.org

D. Demner-Fushman and J. Lin. 2005. Knowledge extraction for clinical question answering: Preliminary results. *AAAI Workshop on Question Answering in Restricted Domains*.

Evidence Based Medicine. Available from: http://ebm.bmjjournals.com

X. Huang et al. 2006. Evaluation of PICO as a Knowledge Representation for Clinical Questions. *Ann. Symp. of AMIA* pp359–363.

L. Hunter and K. Bretonnel Cohen. 2006. Biomedical language processing: what's beyond PubMed? *Molecular Cell*, 21:589-594.

J. Lin et al. 2006. Generative Content Models for Structural Analysis of Medical Abstracts. *Workshop on Biomedical Natural Language Processing BioNLP* New York.

D. A. Lindberg et al. 1993. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291.

A. McCallum. 2002. *MALLET: A Machine Learning for Language Toolkit.* http://mallet.cs.umass.edu.

L. McKnight and P. Srinivasan 2003. Categorization of Sentence Types in Medical Abstracts. *Ann. Symp. of AMIA* pp440–444.

M. Lee et al. 2006. Beyond Information Retrieval–Medical Question Answering. *Ann. Symp. of AMIA.*

Y. Niu and G. Hirst. 2005. Analysis of semantic classes in medical text for question answering. *Workshop on Question Answering in Restricted Domains*, Barcelona.

C. Orasan. 2001. Patterns in Scientific Abstracts. *2001 Corpus Linguistics Conference.*

W. S. Richardson et al. 1995. The well-built clinical question: a key to evidence-based decisions. *ACP J Club*, Nov-Dec;123(3):A12-3.

T. Rindflesch and M. Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: Interpreting hypernymic propositions in biomedical text. *J. of Biomedical Informatics*, 36(6):462–477, Dec.

D. L. Sackett et al.. 1998. *Evidence Based Medicine: How to Practice and Teach EBM.* Churchill Livingstone, Edinburgh.

F. Salager-Meyer. 1990. Discourse Movements in Medical English Abstracts and their linguistic exponents: A genre analysis study. *INTERFACE: J. of Applied Linguistics*, 4(2):107–124.

I. Sim et al. 2000. Electronic Trial Banks: A Complementary Method for Reporting Randomized Trials. *Med Decis Making*, Oct-Dec;20(4):440-50.

C. Sutton and A McCallum. 2006. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning.* MIT Press. To appear.

J. Swales. 1990. *Genre Analysis: English in Academic and Research Settings.* Cambridge University Press, Cambridge University.

I. H. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*, 2nd Ed, Morgan Kaufmann, San Francisco.

R. Xu et al. 2006. Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts. *Ann. Symp. of AMIA.*

P. Zweigenbaum. 2003. Question answering in biomedicine. *Workshop on Natural Language Processing for Question Answering*, Budapest.

Automatic Code Assignment to Medical Text

Koby Crammer and **Mark Dredze** and **Kuzman Ganchev** and **Partha Pratim Talukdar**
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA
{crammer|mdredze|kuzman|partha}@seas.upenn.edu

Steven Carroll
Division of Oncology, The Children's Hospital of Philadelphia, Philadelphia, PA
carroll@genome.chop.edu

Abstract

Code assignment is important for handling large amounts of electronic medical data in the modern hospital. However, only expert annotators with extensive training can assign codes. We present a system for the assignment of ICD-9-CM clinical codes to free text radiology reports. Our system assigns a code configuration, predicting one or more codes for each document. We combine three coding systems into a single learning system for higher accuracy. We compare our system on a real world medical dataset with both human annotators and other automated systems, achieving nearly the maximum score on the Computational Medicine Center's challenge.

1 Introduction

The modern hospital generates tremendous amounts of data: medical records, lab reports, doctor notes, and numerous other sources of information. As hospitals move towards fully electronic record keeping, the volume of this data only increases. While many medical systems encourage the use of structured information, including assigning standardized codes, most medical data, and often times the most important information, is stored as unstructured text.

This daunting amount of medical text creates exciting opportunities for applications of learning methods, such as search, document classification, data mining, information extraction, and relation extraction (Shortliffe and Cimino, 2006). These applications have the potential for considerable benefit to the medical community as they can leverage information collected by hospitals and provide incentives for electronic record storage. Much of the data generated by medical personnel is unused past the clinical visit, often times because there is no way to simply and quickly apply the wealth of information. Medical NLP holds the promise of both greater care for individual patients and enhanced knowledge about health care.

In this work we explore the assignment of ICD-9-CM codes to clinical reports. We focus on this practical problem since it is representative of the type of task faced by medical personnel on a daily basis. Many hospitals organize and code documents for later retrieval using different coding standards. Often times, these standards are extremely complex and only trained expert coders can properly perform the task, making the process of coding documents both expensive and unreliable since a coder must select from thousands of codes a small number for a given report. An accurate automated system would reduce costs, simplify the task for coders, and create a greater consensus and standardization of hospital data.

This paper addresses some of the challenges associated with ICD-9-CM code assignment to clinical free text, as well as general issues facing applications of NLP to medical text. We present our automated system for code assignment developed for the Computational Medicine Center's challenge. Our approach uses several classification systems, each with the goal of predicting the exact code configuration for a medical report. We then use a learning

BioNLP 2007: Biological, translational, and clinical language processing, pages 129–136,
Prague, June 2007.

system to combine our predictions for superior performance.

This paper is organized as follows. First, we explain our task and difficulties in detail. Next we describe our three automated systems and features. We combine the three approaches to create a single superior system. We evaluate our system on clinical reports and show accuracy approaching human performance and the challenge's best score.

2 Task Overview

The health care system employs a large number of categorization and classification systems to assist data management for a variety of tasks, including patient care, record storage and retrieval, statistical analysis, insurance, and billing. One of these systems is the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) which is the official system of assigning codes to diagnoses and procedures associated with hospital utilization in the United States. [1] The coding system is based on World Health Organization guidelines. An ICD-9-CM code indicates a classification of a disease, symptom, procedure, injury, or information from the personal history. Codes are organized hierarchically, where top level entries are general groupings (e.g. "diseases of the respiratory system") and bottom level codes indicate specific symptoms or diseases and their location (e.g. "pneumonia in aspergillosis"). Each specific, low-level code consists of 4 or 5 digits, with a decimal after the third. Higher level codes typically include only 3 digits. Overall, there are thousands of codes that cover a broad range of medical conditions.

Codes are assigned to medical reports by doctors, nurses and other trained experts based on complex coding guidelines (National Center for Health Statistics, 2006). A particular medical report can be assigned any number of relevant codes. For example, if a patient exhibits a cough, fever and wheezing, all three codes should be assigned. In addition to finding appropriate codes for each condition, complex rules guide code assignment. For example, a diagnosis code should always be assigned if a diagnosis is reached, a diagnosis code should never be assigned when the diagnosis is unclear, a symptom should never be assigned when a diagnosis is present, and the most specific code is preferred. This means that codes that seem appropriate to a report should be omitted in specific cases. For example, a patient with hallucinations should be coded 780.1 (hallucinations) but for visual hallucinations, the correct code is 368.16. The large number of codes and complexity of assignment rules make this a difficult problem for humans (inter-annotator agreement is low). Therefore, an automated system that suggested or assigned codes could make medical data more consistent.

These complexities make the problem difficult for NLP systems. Consider the task as multi-class, multi-label. For a given document, many codes may seem appropriate but it may not be clear to the algorithm how many to assign. Furthermore, the codes are not independent and different labels can interact to either increase or decrease the likelihood of the other. Consider a report that says, "patient reports cough and fever." The presence of the words cough and fever indicate codes 786.2 (cough) and 780.6 (fever). However, if the report continues to state that "patient has pneumonia" then these codes are dropped in favor of 486 (pneumonia). Furthermore, if the report then says "verify clinically", then the diagnosis is uncertain and only codes 786.2 and 780.6 apply. Clearly, this is a challenging problem, especially for an automated system.

2.1 Corpus

We built and evaluated our system in accordance with the Computational Medicine Center's (CMC) 2007 Medical Natural Language Processing Challenge.[2] Since release of medical data must strictly follow HIPAA standards, the challenge corpus underwent extensive treatment for disambiguation, anonymization, and careful scrubbing. A detailed description of data preparation is found in Computational Medicine Center (2007). We describe the corpus here to provide context for our task.

The training corpus is comprised of 978 radiological reports taken from real medical records. A test corpus contains 976 unlabeled documents. Radiology reports have two text fields, clinical history and

[1] http://www.cdc.gov/nchs/about/otheract/icd9/abticd9.htm

[2] www.computationalmedicine.org/challenge

impression. The physician ordering the x-ray writes the clinical history, which contains patient information for the radiologist, including history and current symptoms. Sometimes a guess as to the diagnosis appears ("evaluate for asthma"). The descriptions are sometimes whole sentences and other times single words ("cough"). The radiologist writes the impression to summarize his or her findings. It contains a short analysis and often times a best guess as to the diagnosis. At times this field is terse, ("pneumonia" or "normal kidneys") and at others it contains an entire paragraph of text. Together, these two fields are used to assign ICD-9-CM codes, which justify a certain procedure, possibly for reimbursement by the insurance company.

Only a small percentage of ICD-9-CM codes appear in the challenge. In total, the reports include 45 different codes arranged in 94 configurations (combinations). Some of these codes appear frequently, while others are rare, appearing only a single time. The test set is restricted so that each configuration appears at least once in the training set, although there is no further guarantee as to the test set's distribution over codes. Therefore, in addition to a large number of codes, there is variability in the amount of data for each code. Four codes have over 100 examples each and 24 codes have 10 or fewer documents, with 10 of these codes having only a single document.

Since code annotation is a difficult task, each document in the corpus was evaluated by three expert annotators. A gold annotation was created by taking the majority of the annotators; if two of the three annotators provided a code, that code is used in the gold configuration. This approach means that a document's configuration may be a construction of multiple annotators and may not match any of the three annotators exactly. Both the individual and the majority annotations are included with the training corpus.

While others have attempted ICD-9 code classification, our task differs in two respects (Section 7 provides an overview of previous work). First, previous work has used discharge reports, which are typically longer with more text fields. Second, while most systems are evaluated as a recommendation system, offering the top k codes and then scoring recall at k, our task is to provide the *exact* configuration. The CMC challenge evaluated systems using an F1 score, so we are penalized if we suggest any label that does not appear in the majority annotation.

To estimate task difficulty we measured the inter-annotator score for the training set using the three annotations provided. We scored two annotations with the micro average F1, which weighs each code assignment equally (see Section 5 for details on evaluation metrics). If an annotator omitted a code and included an extra code, he or she is penalized with a false positive (omitting a code) and a false negative (adding an extra code). We measured annotators against each other; the average f-measure was 74.85 (standard deviation of .06). These scores were low since annotators chose from an unrestricted set of codes, many of which were not included in the final majority annotation. However, these scores still indicate the human accuracy for this task using an unrestricted label set. [3]

3 Code Assignment System

We developed three automated systems guided by our above analysis. First, we designed a learning system that used natural language features from the official code descriptions and the text of each report. It is general purpose and labels all 45 codes and 94 configurations (labels). Second, we built a rule based system that assigned codes based on the overlap between the reports and code descriptions, similar to how an annotator may search code descriptions for appropriate labels. Finally, a specialized system aimed at the most common codes implemented a policy that mimics the guidelines a medical staffer would use to assign these codes.

3.1 Learning System

We begin with some notational definitions. In what follows, x denotes the generic input document (radiology report), Y denotes the set of possible labelings (code configurations) of x, and $y^*(x)$ the correct labeling of x. For each pair of document x and labeling $y \in Y$, we compute a vector-valued feature representation $f(x, y)$. A linear model is

[3] We also measured each annotator with the majority codes, taking the average score (87.48), and the best annotator with the majority label (92.8). However, these numbers are highly biased since the annotator influences the majority labeling. We observe that our final system still exceeds the average score.

given by a weight vector w. Given this weight vector w, the score $w \cdot f(x, y)$ ranks possible labelings of x, and we denote by $Y_{k,w}(x)$ the set of k top scoring labelings for x. For some structured problems, a factorization of $f(x, y)$ is required to enable a dynamic program for inference. For our problem, we know all the possible configurations in advance (there are 94 of them) so we can pick the highest scoring $y \in Y$ by trying them all. For each document x and possible labeling y, we compute a score using w and the feature representation $f(x, y)$. The top scoring y is output as the correct label. Section 3.1.1 describes our feature function $f(x, y)$ while Section 3.1.2 describes how we find a good weight vector w.

3.1.1 Features

Problem representation is one of the most important aspects of a learning system. In our case, this is defined by the set of features $f(x, y)$. Ideally we would like a linear combination of our features to exactly specify the true labeling of all the instances, but we want to have a small total number of features so that we can accurately estimate their values. We separate our features into two classes: label specific features and transfer features. For simplicity, we index features by their name. Label specific features are only present for a single label. For example, a simple class of label specific features is the conjunction of a word in the document with an ICD-9-CM code in the label. Thus, for each word we create 94 features, i.e. the word conjoined with every label. These features tend to be very powerful, since weights for them can encode very specific information about the way doctors talk about a disease, such as the feature "contains word pneumonia and label contains code 486". Unfortunately, the cost of this power is that there are a large number of these features, making parameter estimation difficult for rare labels. In contrast, transfer features can be present in multiple labels. An example of a transfer feature might be "the impression contains all the words in the code descriptions of the codes in this label". Transfer features allow us to generalize from one label to another by learning things like "if all the words of the label description occur in the impression, then this label is likely" but have the drawback that we cannot learn specific details about common labels. For example, we cannot learn that the word "pneumonia" in the impression is negatively correlated with the code cough. The inclusion of both label specific and transfer features allows us to learn specificity where we have a large number of examples and generality for rare codes.

Before feature extraction we normalized the reports' text by converting it to lower case and by replacing all numbers (and digit sequences) with a single token $\langle NUM \rangle$. We also prepared a synonym dictionary for a subset of the tokens and n-grams present in the training data. The synonym dictionary was based on MeSH[4], the Medical Subject Headings vocabulary, in which synonyms are listed as terms under the same concept. All ngrams and tokens in the training data which had mappings defined in the synonym dictionary were then replaced by their normalized token; e.g. all mentions of "nocturnal enuresis" or "nighttime urinary incontinence" were replaced by the token "bedwetting". Additionally, we constructed descriptions for each code automatically from the official ICD-9-CM code descriptions in National Center for Health Statistics (2006). We also created a mapping between code and code type (diagnosis or symptom) using the guidelines.

Our system used the following features. The descriptions of particular features are in quotes, while schemes for constructing features are not.

- "this configuration contains a disease code", "this configuration contains a symptom code", "this configuration contains an ambiguous code" and "this configuration contains both disease and symptom codes".[5]

- With the exception of stop-words, all words of the impression and history conjoined with each label in the configuration; pairs of words conjoined with each label; words conjoined with pairs of labels. For example, "the impression contains 'pneumonia' and the label contains codes 786.2 and 780.6".

- A feature indicating when the history or impression contains a complete code description

[4] www.nlm.nih.gov/mesh

[5] We included a feature for configurations that had both disease and symptom codes because they appeared in the training data, even though coding guidelines prohibit these configurations.

for the label; one for a word in common with the code description for one of the codes in the label; a common word conjoined with the presence of a negation word nearby ("no", "not", etc.); a word in common with a code description not present in the label. We applied similar features using negative words associated with each code.

- A feature indicating when a soft negation word appears in the text ("probable", "possible", "suspected", etc.) conjoined with words that follow; the token length of a text field ("impression length=3"); a conjunction of a feature indicating a short text field with the words in the field ("impression length=1 and 'pneumonia' ")

- A feature indicating each n-gram sequence that appears in both the impression and clinical history; the conjunction of certain terms where one appears in the history and the other in the impression (e.g. "cough in history and pneumonia in impression").

3.1.2 Learning Technique

Using these feature representations, we now learn a weight vector w that scores the correct labelings of the data higher than incorrect labelings. We used a k-best version of the MIRA algorithm (Crammer, 2004; McDonald et al., 2005). MIRA is an online learning algorithm that for each training document x updates the weight vector w according to the rule:

$$\begin{aligned} w_{\text{new}} &= \arg\min_{w} \|w - w_{\text{old}}\| \\ &\text{s.t.} \quad \forall y \in Y_{k,w_{old}}(x) : \\ w \cdot f(x, y^*(x)) - w \cdot f(x, y) &\geq L(y^*(x), y) \end{aligned}$$

where $L(y^*(x), y)$ is a measure of the loss of labeling y with respect to the correct labeling $y^*(x)$. For our experiments, we set k to 30 and iterated over the training data 10 times. Two standard modifications to this approach also helped. First, rather than using just the final weight vector, we average all weight vectors. This has a smoothing effect that improves performance on most problems. The second modification is the introduction of slack variables:

$$\begin{aligned} w_{\text{new}} &= \arg\min_{w} \|w - w_{\text{old}}\| + \gamma \sum_i \xi_i \\ &\text{s.t.} \quad \forall y \in Y_{k,w_{old}}(x) : \\ w \cdot f(x, y^*(x)) - w \cdot f(x, y) &\geq L(y^*(x), y) - \xi_i \\ &\forall i \in \{1 \ldots k\} : \quad \xi_i \geq 0. \end{aligned}$$

We used a γ of 10^{-3} in our experiments.

The most straightforward loss function is the 0/1 loss, which is one if y does not equal $y^*(x)$ and zero otherwise. Since we are evaluated based on the number of false negative and false positive ICD-9-CM codes assigned to all the documents, we used a loss that is the sum of the number of false positive and the number of false negative labels that y assigns with respect to $y^*(x)$.

Finally, we only used features that were possible for some labeling of the test data by using only the test data to construct our feature alphabet. This forced the learner to focus on hypotheses that could be used at test time and resulted in a 1% increase in F-measure in our final system on the test data.

3.2 Rule Based System

Since some of the configurations appear a small number of times in our corpus (some only once), we built a rule based system that requires no training. The system uses a description of the ICD-9-CM codes and their types, similar to the list used by our learning system (Section 3.1.1). The code descriptions include between one and four short descriptions, such as "reactive airway disease", "asthma", and "chronic obstructive pulmonary disease". We treat each of these descriptions as a bag of words. For a given report, the system parses both the clinical history and impression into sentences, using "." as a sentence divider. Each sentence is the checked to see if all of the words in a code description appear in the sentence. If a match is found, we set a flag corresponding to the code. However, if the code is a disease, we search for a negation word in the sentence, removing the flag if a negation word is found. Once all code descriptions have been evaluated, we check if there are any flags set for disease codes. If so, we remove all symptom code flags. We then emit a code corresponding to each set flag. This simple system does not enforce configuration restrictions;

we may predict a code configuration that does not appear in our training data. Adding this restriction improved precision but hurt recall, leading to a slight decrease in F1 score. We therefore omitted the restriction from our system.

3.3 Automatic Coding Policies

As we described in Section 2, enforcing coding guidelines can be a complex task. While a learning system may have trouble coding a document, a human may be able to define a simple policy for coding. Since some of the most frequent codes in our dataset have this property, we decided to implement such an automatic coding policy. We selected two related sets of codes to target with a rule based system, a set of codes found in pneumonia reports and a set for urinary tract infection/reflux reports.

Reports related to pneumonia are the most common in our dataset and include codes for pneumonia, asthma, fever, cough and wheezing; we handle them with a single policy. Our policy is as follows:

- Search for a small set of keywords (e.g. "cough", "fever") to determine if a code should be applied.
- If "pneumonia" appears unnegated in the impression and the impression is short, or if it occurs in the clinical history and is not preceded by phrases such as "evaluate for" or "history of", apply pneumonia code and stop.
- Use the same rule to code asthma by looking for "asthma" or "reactive airway disease".
- If no diagnosis is found, code all non-negated symptoms (cough, fever, wheezing).

We selected 80% of the training set to evaluate in the construction of our rules. We then ran the finished system on both this training set and the held out 20% of the data. The system achieved F1 scores of 87% on the training set and 84% on the held out data for these five codes. The comparable scores indicates that we did not over-fit the training data.

We designed a similar policy for two other related codes, urinary tract infection and vesicoureteral reflux. We found these codes to be more complex as they included a wide range of kidney disorders. On these two codes, our system achieved 78% on the train set and 76% on the held out data. Overall, automatically applying our two policies yielded high confidence predictions for a significant subset of the corpus.

4 Combined System

Since our three systems take complimentary approaches to the problem, we combined them to improve performance. First, we took our automatic policy and rule based systems and cascaded them; if the automatic policy system does not apply a code, the rule based system classifies the report. We used a cascaded approach since the automatic policy system was very accurate when it was able to assign a code. Therefore, the rule based system defers to the policy system when it is triggered. Next, we included the prediction of the cascaded system as a feature for our learning system. We used two feature rules: "cascaded-system predicted exactly this label" and "cascaded-system predicted one of the codes in this label". As we show, this yielded our most accurate system. While we could have used a meta-classifier to combine the three systems, including the rule based systems as features to the learning system allowed it to learn the appropriate weights for the rule based predictions.

5 Evaluation Metric

Evaluation metrics for this task are often based on recommendation systems, where the system returns a list of the top k codes for selection by the user. As a result, typical metrics are "recall at k" and average precision (Larkey and Croft, 1995). Instead, our goal was to predict the exact configuration, returning exactly the number of codes predicted to be on the report. The competition used a micro-averaged F1 score to evaluate predictions. A contingency table (confusion matrix) is computed by summing over each predicted code for each document by prediction type (true positive, false positive, false negative) weighing each code assignment equally. F1 score is computed based on the resultant table. If specific codes or under-coding is favored, we can modify our learning loss function as described in Section 3.1.2. A detailed treatment of this evaluation metric can be found in Computational Medicine Center (2007).

System	*Precision*	*Recall*	*F1*
BL	61.86	72.58	66.79
RULE	81.9	82.0	82.0
CASCADE	86.04	84.56	85.3
LEARN	85.5	83.6	84.6
CASCADE+LEARN	87.1	85.9	86.5

Table 1: Performance of our systems on the provided labeled training data (F1 score). The learning systems (*CASCADE+LEARN* and *LEARN*) were evaluated on ten random split of the data while *RULE* was evaluated on all of the training data. We include a simple rule based system (*BL*) as a baseline.

6 Results

We evaluated our systems on the labeled training data of 978 radiology reports. For each report, each system predicted an exact configuration of codes (i.e. one of 94 possible labels). We score each system using a micro-averaged F1 score. Since we only had labels for the training data, we divided the data using an 80/20 training test split and averaged results over 10 runs for our learning systems. We evaluated the following systems:

- *RULE* : The rule based system based on ICD-9-CM code descriptions (Section 3.2).
- *CASCADE* : The automatic code policy system (Section 3.3) cascaded with *RULE* (Section 4).
- *LEARN* : The learning system with both label specific and transfer features (Section 3.1).
- *CASCADE+LEARN* : Our combined system that incorporates *CASCADE* predictions as a feature to *LEARN* (Section 4).

For a baseline, we built a simple system that applies the official ICD-9-CM code descriptions to find the correct labels (*BL*). For each code in the training set, the system generates text-segments related to it. During testing, for each new document, the system checks if any text-segment (as discovered during training) appears in the document. If so, the corresponding code is predicted. The results from our four systems and baseline are shown in Table 1.

System	*Train*	*Test*
CASCADE	85.3	84
CASCADE+LEARN	86.5	87.60
Average	-	76.6
Best	-	89.08

Table 2: Performance of two systems on the train and test data. Results obtained from the web submission interface were rounded. *Average* and *Best* are the average and best f-measures of the 44 submitted systems (standard deviation 13.40).

Each of our systems easily beats the baseline, and the average inter-annotator score for this task. Additionally, we were able to evaluate two of our systems on the test data using a web interface as provided by the competition. The test set contains 976 documents (about the same as the training set) and is drawn the from same distribution as the training data. Our test results were comparable to performance on the training data, showing that we did not over-fit to the training data (Table 2). Additionally, our combined system (*CASCADE+LEARN*) achieved a score of 87.60%, beating our training data performance and exceeding the average inter-annotator score. Out of 44 submitted systems, the average score on test data was 76.7% (standard deviation of 13.40) and the maximum score was 89.08%. Our system scored 4th overall and was less than 1.5% behind the best system. Overall, in comparison with our baselines and over 40 systems, we perform very well on this task.

7 Related Work

There have been several attempts at ICD-9-CM code classification and related problems for medical records. The specific problem of ICD-9-CM code assignment was studied by Lussier et al. (2000) through an exploratory study. Larkey and Croft (1995) designed classifiers for the automatic assignment of ICD-9 codes to discharge summaries. Discharge summaries tend to be considerably longer than our data and contain multiple text fields. Additionally, the number of codes per document has a larger range, varying between 1 and 15 codes. Larkey and Croft use three classifiers: K-nearest neighbors, relevance feedback, and bayesian inde-

pendence. Similar to our approach, they tag items as negated and try to identify diagnosis and symptom terms. Additionally, their final system combines all three models. A direct comparison is not possible due to the difference in data and evaluation metrics; they use average precision and recall at k. On a comparable metric, "principal code is top candidate", their best system achieves 59.9% accuracy. de Lima et al. (1998) rely on the hierarchical nature of medical codes to design a hierarchical classification scheme. This approach is likely to help on our task as well but we were unable to test this since the limited number of codes removes any hierarchy. Other approaches have used a variety of NLP techniques (Satomura and Amaral, 1992).

Others have used natural language systems for the analysis of medical records (Zweigenbaum, 1994). Chapman and Haug (1999) studied radiology reports looking for cases of pneumonia, a goal similar to that of our automatic coding policy system. Meystre and Haug (2005) processed medical records to harvest potential entries for a medical problem list, an important part of electronic medical records. Chuang et al. (2002) studied Charlson comorbidities derived from processing discharge reports and chest x-ray reports and compared them with administrative data. Additionally, Friedman et al. (1994) applies NLP techniques to radiology reports.

8 Conclusion

We have presented a learning system that processes radiology reports and assigns ICD-9-CM codes. Each of our systems achieves results comparable with an inter-annotator baseline for our training data. A combined system improves over each individual system. Finally, we show that on test data unavailable during system development, our final system continues to perform well, exceeding the inter-annotator baseline and achieving the 4th best score out of 44 systems entered in the CMC challenge.

9 Acknowledgements

We thank Andrew Lippa for his extensive medical wisdom. Dredze is supported by an NDSEG fellowship; Ganchev and Talukdar by NSF ITR EIA-0205448; and Crammer by DARPA under Contract No. NBCHD03001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

References

W.W. Chapman and P.J. Haug. 1999. Comparing expert systems for identifying chest x-ray reports that support pneumonia. In *AMIA Symposium*, pages 216–20.

JH Chuang, C Friedman, and G Hripcsak. 2002. A comparison of the charlson comorbidities derived from medical language processing and administrative data. *AMIA Symposium*, pages 160–4.

Computational Medicine Center. 2007. The computational medicine center's 2007 medical natural language processing challenge. http://computationalmedicine.org/challenge/index.php.

Koby Crammer. 2004. *Online Learning of Complex Categorial Problems*. Ph.D. thesis, Hebrew Univeristy of Jerusalem.

Luciano R. S. de Lima, Alberto H. F. Laender, and Berthier A. Ribeiro-Neto. 1998. A hierarchical approach to the automatic categorization of medical documents. In *CIKM*.

C Friedman, PO Alderson, JH Austin, JJ Cimino, and SB Johnson. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1:161–74.

Leah S. Larkey and W. Bruce Croft. 1995. Automatic assignment of icd9 codes to discharge summaries. Technical report, University of Massachusetts at Amherst, Amherst, MA.

YA Lussier, C Friedman, L Shagina, and P Eng. 2000. Automating icd-9-cm encoding using medical language processing: A feasibility study.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Flexible text segmentation with structured multilabel classification. In *HLT/EMNLP*.

Stephane Meystre and Peter J Haug. 2005. Automation of a problem list using natural language processing. *BMC Medical Informatics and Decision Making*.

National Center for Health Statistics. 2006. Icd-9-cm official guidelines for coding and reporting. http://www.cdc.gov/nchs/datawh/ftpserv/ftpicd9/ftpicd9.htm.

Y Satomura and MB Amaral. 1992. Automated diagnostic indexing by natural language processing. *Medical Informatics*, 17:149–163.

Edward H. Shortliffe and James J. Cimino, editors. 2006. *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. Springer.

P. Zweigenbaum. 1994. Menelas: an access system for medical records using natural language. *Comput Methods Programs Biomed*, 45:117–20.

Interpreting Comparative Constructions in Biomedical Text

Marcelo Fiszman,[1] **Dina Demner-Fushman,**[2]
Francois M. Lang,[2] **Philip Goetz,**[2]
Thomas C. Rindflesch[2]
[1]University of Tennessee – GSM, Knoxville, TN 37920
mfiszman@utmck.edu
[2]Lister Hill National Center for Biomedical Communications
National Library of Medicine, Bethesda, MD 20894
{ddemner|goetzp|flang|trindflesch}@mail.nih.gov

Abstract

We propose a methodology using underspecified semantic interpretation to process comparative constructions in MEDLINE citations, concentrating on two structures that are prevalent in the research literature reporting on clinical trials for drug therapies. The method exploits an existing semantic processor, SemRep, which constructs predications based on the Unified Medical Language System. Results of a preliminary evaluation were recall of 70%, precision of 96%, and F-score of 81%. We discuss the generalization of the methodology to other entities such as therapeutic and diagnostic procedures. The available structures in computable format are potentially useful for interpreting outcome statements in MEDLINE citations.

1 Introduction

As natural language processing (NLP) is increasingly able to support advanced information management techniques for research in medicine and biology, it is being incrementally improved to provide extended coverage and more accurate results. In this paper, we discuss the extension of an existing semantic interpretation system to address comparative structures. These structures provide a way of explicating the characteristics of one entity in terms of a second, thereby enhancing the description of the first. This phenomenon is important in clinical research literature reporting the results of clinical trials.

In the abstracts of these reports, a treatment for some disease is typically discussed using two types of comparative structures. The first announces that the (primary) therapy focused on in the study will be compared to some other (secondary) therapy. A typical example is (1).

(1) Lansoprazole compared with ranitidine for the treatment of nonerosive gastroesophageal reflux disease.

An outcome statement (2) often appears near the end of the abstract, asserting results in terms of the relative merits of the primary therapy compared to the secondary.

(2) Lansoprazole is more effective than ranitidine in patients with endoscopically confirmed non-erosive reflux esophagitis.

The processing of comparative expressions such as (1) and (2) was incorporated into an existing system, SemRep [Rindflesch and Fiszman, 2003; Rindflesch et al., 2005], which constructs semantic predications by mapping assertions in biomedical text to the Unified Medical Language System® (UMLS)® [Humphreys et al., 1998].

2 Background

2.1 Comparative structures in English

The range of comparative expressions in English is extensive and complex. Several linguistic studies have investigated their characteristics, with differing assumptions about syntax and semantics (for example [Ryan, 1981; Rayner and Banks, 1990; Staab and Hahn, 1997; Huddleston and Pullum, 2002]). Our study concentrates on

BioNLP 2007: Biological, translational, and clinical language processing, pages 137–144,
Prague, June 2007.

structures in which two drugs are compared with respect to a shared attribute (e.g. how well they treat some disease). An assessment of their relative merit in this regard is indicated by their positions on a scale. The compared terms are expressed as noun phrases, which can be considered to be conjoined. The shared characteristic focused on is expressed as a predicate outside the comparative structure. An adjective or noun is used to denote the scale, and words such as *than, as, with,* and *to* serve as cues to identify the compared terms, the scale, and the relative position of the terms on the scale.

The first type of structure we address (called comp1 and illustrated in (3)) merely asserts that the primary and secondary terms (in bold) are being compared. A possible cue for identifying these structures is a form of *compare*. A further characteristic is that the compared terms are separated by a conjunction, or a preposition, as in (3).

(3) To compare **misoprostol** with **dinoprostone** for cervical ripening and labor induction.

As shown in (4), a scale may be mentioned (*efficacy*); however, in this study, we only identify the compared terms in structures of this type.

(4) To compare the *efficacy* of **misoprostol** with **dinoprostone** for cervical ripening and labor induction.

In the more complex comparative expression we accommodate (called comp2), the relative ranking of two compared terms is indicated on a scale denoted by an adjective (e.g. *effective* in (5)). The relative position of the compared terms in scalar comparative structures of this type expresses either equality or inequality. Inequality is further divided into superiority, where the primary compared term is higher on the scale than the secondary, and inferiority, where the opposite is true. Cues associated with the adjective designating the scale signal these phenomena (e.g. *as* ADJ *as* in (5) for equality, ADJer *than* in (6) for superiority, and *less* ADJ *than* in (7) for inferiority).

(5) **Azithromycin** is as effective as **erythromycin estolate** for the treatment of pertussis in children.

(6) **Naproxen** is safer than **aspirin** in the treatment of the arthritis of rheumatic fever.

(7) **Sodium valproate** was significantly less effective than **prochlorperazine** in reducing pain or nausea.

In examples (3) through (7), the characteristic the compared drugs have in common is treatment of some disorder, for example *treatment of pertussis in children* in (5).

Few studies describe an implemented automatic analysis of comparatives; however, Friedman [Friedman, 1989] is a notable exception. Jindal and Liu [Jindal and Liu, 2006] use machine learning to identify some comparative structures, but do not provide a semantic interpretation. We exploit SemRep machinery to interpret the aspects of comparative structures just described.

2.2 SemRep

SemRep [Rindflesch and Fiszman, 2003; Rindflesch et al., 2005] recovers underspecified semantic propositions in biomedical text based on a partial syntactic analysis and structured domain knowledge from the UMLS. Several systems that extract entities and relations are under development in both the clinical and molecular biology domains. Examples of systems for clinical text are described in [Friedman et al., 1994], [Johnson et al., 1993], [Hahn et al., 2002], and [Christensen et al., 2002]. In molecular biology, examples include [Yen et al., 2006], [Chun et al., 2006], [Blaschke et al., 1999], [Leroy et al., 2003], [Rindflesch et al., 2005], [Friedman et al., 2001], and [Lussier et al., 2006].

During SemRep processing, a partial syntactic parse is produced that depends on lexical look-up in the SPECIALIST lexicon [McCray et al., 1994] and a part-of-speech tagger [Smith et al., 2004]. MetaMap [Aronson, 2001] then matches noun phrases to concepts in the Metathesaurus® and determines the semantic type for each concept. For example, the structure in (9), produced for (8), allows both syntactic and semantic information to be used in further SemRep processing that interprets semantic predications.

(8) Lansoprazole for the treatment of gastroesophageal reflux disease

```
(9) [[head(noun(Lansoprazole),metaconc('lansoprazole':[phsu]))],[prep(for),det(the),head(noun(treatment))],[prep(of),mod(adj(gastroesophageal)),mod(noun(reflux)),head(noun(disease),metaconc('Gastroesophageal reflux disease':[dsyn]))]]
```

Predicates are derived from indicator rules that map syntactic phenomena (such as verbs and nominalizations) to relationships in the UMLS Semantic Network. Argument identification is guided by dependency grammar rules as well as constraints imposed by the Semantic Network. In processing (8), for example, an indicator rule links the nominalization *treatment* with the Semantic Network relation "Pharmacologic Substance TREATS Disease or Syndrome." Since the semantic types of the syntactic arguments identified for *treatment* in this sentence ('Pharmacologic Substance' for "lansoprazole" and 'Disease or Syndrome' for "Gastroesophageal reflux disease") match the corresponding semantic types in the relation from the Semantic Network, the predication in (10) is constructed, where subject and object are Metathesaurus concepts.

```
(10) lansoprazole TREATS Gastroesophageal reflux disease
```

3 Methods

3.1 Linguistic patterns

We extracted sentences for developing comparative processing from a set of some 10,000 MEDLINE citations reporting on the results of clinical trials, a rich source of comparative structures. In this sample, the most frequent patterns for comp1 (only announces that two terms are compared) and comp2 (includes a scale and positions on that scale) are given in (11) and (12). In the patterns, Term1 and Term2 refer to the primary and secondary compared terms, respectively. "{BE}" means that some form of *be* is optional, and slash indicates disjunction. These patterns served as guides for enhancing SemRep argument identification machinery but were not implemented as such. That is, they indicate necessary components but do not preclude intervening modifiers and qualifiers.

(11) **comp1**: Compared terms

C1: Term1 {BE} *compare with*/*to* Term2
C2: *compare* Term1 *with*/*to* Term2
C3: *compare* Term1 *and*/*versus* Term2
C4a: Term1 *comparison with*/*to* Term2
C4b: *comparison of* Term1 *with*/*to* Term2
C4c: *comparison of* Term1 *and*/*versus* Term2
C5 Term1 *versus* Term2

(12) **comp2**: Scalar patterns

S1: Term1 BE *as* ADJ *as* {BE} Term2
S2a: Term1 BE *more* ADJ *than* {BE} Term2
S2b: Term1 BE ADJer *than* {BE}Term2
S2c: Term1 BE *less* ADJ *than* {BE} Term2
S4: Term1 BE *superior to* Term2
S5: Term1 BE *inferior to* Term2

As with SemRep in general, the interpretation of comparative structures exploits underspecified syntactic structure enhanced with Metathesaurus concepts and semantic types. Semantic groups [McCray et al., 2001] from the Semantic Network are also available. For this project, we exploit the group Chemicals & Drugs, which contains such semantic types as 'Pharmacologic Substance', 'Antibiotic', and 'Immunologic Factor'. (The principles used here also apply to compared terms with semantic types from other semantic groups, such as 'Procedures'.) In the comp1 patterns, a form of *compare* acts as an indicator of a comparative predication. In comp2, the adjective serves that function. Other words appearing in the patterns cue the indicator word (in comp2) and help identify the compared terms (in both comp1 and comp2). The conjunction v*ersus* is special in that it cues the secondary compared term (Term2) in comp1, but may also indicate a comp1 structure in the absence of a form of *compare* (C5).

3.2 Interpreting comp1 patterns

When SemRep encounters a form of *compare*, it assumes a comp1 structure and looks to the right for the first noun phrase immediately preceded by *with*, *to*, *and*, or *versus*. If the head of this phrase is mapped to a concept having a semantic type in the group Chemicals & Drugs, it is marked as the secondary compared term. The algorithm then looks to the left of that term for a noun phrase having a semantic type also in the group Chemicals & Drugs, which becomes the primary compared term. When this processing is applied to (13), the semantic predication (14) is produced, in which the predicate is COMPARED_WITH; the first argument is the primary compared term and the

other is the secondary. As noted earlier, although a scale is sometimes asserted in these structures (as in (13)), SemRep does not retrieve it. An assertion regarding position on the scale never appears in comp1 structures.

(13) To compare the efficacy and tolerability of **Hypericum perforatum** with **imipramine** in patients with mild to moderate depression.

(14) Hypericum perforatum COMPARED_WITH Imipramine

SemRep considers noun phrases occurring immediately to the right and left of *versus* as being compared terms if their heads have been mapped to Metathesaurus concepts having semantic types belonging to the group Chemicals & Drugs. Such noun phrases are interpreted as part of a comp1 structure, even if a form of *compare* has not occurred. The predication (16) is derived from (15).

(15) Intravenous **lorazepam** versus **dimenhydrinate** for treatment of vertigo in the emergency department: a randomized clinical trial.

(16) Lorazepam COMPARED_WITH Dimenhydrinate

SemRep treats compared terms as being coordinated. For example, this identification allows both "Lorazepam" and "Dimenhydrinate" to function as arguments of TREATS in (15). Consequently, in addition to (16), the predications in (17) are returned as the semantic interpretation of (15). Such processing is done for all comp1 and comp2 structures (although these results are not given for (13) and are not further discussed in this paper).

(17) Lorazepam TREATS Vertigo
Dimenhydrinate TREATS Vertigo

3.3 Interpreting comp2 patterns

In addition to identifying two compared terms when processing comp2 patterns, a scale must be named and the relative position of the terms on that scale indicated. The algorithm for finding compared terms in comp2 structures begins by locating one of the cues *as*, *than*, or *to* and then examines the next noun phrase to the right. If its head has been mapped to a concept with a semantic type in the group Chemicals & Drugs, it is marked as the secondary compared term. As in comp1, the algorithm then looks to the left for the first noun phrase having a head in the same semantic group, and that phrase is marked as the primary compared term.

To find the scale name, SemRep examines the secondary compared term and then locates the first adjective to its left. The nominalization of that adjective (as found in the SPECIALIST Lexicon) is designated as the scale and serves as an argument of the predicate SCALE in the interpretation. For adjectives *superior* and *inferior* (patterns S4 and S5 in (12)) the scale name is "goodness."

In determining relative position on the scale, equality is contrasted with inequality. If the adjective of the construction is immediately preceded by *as* (pattern S1 in (12) above), the two compared terms have the same position on the scale (equality), and are construed as arguments of a predication with predicate SAME_AS. In all other comp2 constructions, the compared terms are in a relationship of inequality. The primary compared term is considered higher on the scale unless the adjective is *inferior* or is preceded by *less*, in which case the secondary term is higher. The predicates HIGHER_THAN and LOWER_THAN are used to construct predications with the compared terms to interpret position on the scale. The equality construction in (18) is expressed as the predications in (19).

(18) **Candesartan** is as effective as **lisinopril** once daily in reducing blood pressure.

(19) Candesartan COMPARED_WITH lisinopril
SCALE:Effectiveness
Candesartan SAME_AS lisinopril

The superiority construction in (20) is expressed as the predications in (21).

(20) **Losartan** was more effective than **atenolol** in reducing cardiovascular morbidity and mortality in patients with hypertension, diabetes, and LVH.

(21) Losartan COMPARED_WITH Atenolol

SCALE:Effectiveness
Losartan HIGHER_THAN Atenolol

The inferiority construction in (22) is expressed as the predications in (23).

(22) **Morphine-6-glucoronide** was significantly less potent than **morphine** in producing pupil constriction.

(23) morphine-6-glucoronide COMPARED_WITH Morphine
SCALE:Potency
morphine-6-glucoronide LOWER_THAN Morphine

3.4 Accommodating negation

Negation in comparative structures affects the position of the compared terms on the scale, and is accommodated differently for equality and for inequality. When a scalar comparison of equality (pattern S1, *as* ADJ *as*) is negated, the primary term is lower on the scale than the secondary (rather than being at least equal). For example, in interpreting the negated equality construction in (24), SemRep produces (25).

(24) **Amoxicillin-clavulanate** was *not* as effective as **ciprofloxacin** for treating uncomplicated bladder infection in women.

(25) Amoxicillin-clavulanate COMPARED_WITH Ciprofloxaci
SCALE:Effectiveness
Amoxicillin-clavulanate LOWER_THAN Ciprofloxacin

For patterns of inequality, SemRep negates the predication indicating position on the scale. For example, the predications in (27) represent the negated superiority comparison in (26). Negation of inferiority comparatives (e.g. "X is not less effective than Y") is extremely rare in our sample.

(26) These data show that **celecoxib** is *not* better than diclofenac (P = 0.414) in terms of ulcer complications.

(27) celecoxib COMPARED_WITH diclofenac
SCALE:Goodness
celecoxib NEG_HIGHER_THAN diclofenac

3.5 Evaluation

To evaluate the effectiveness of the developed methods we created a test set of 300 sentences containing comparative structures. These were extracted by the second author (who did not participate in the development of the methodology) from 3000 MEDLINE citations published later in date than the citations used to develop the methodology. The citations were retrieved with a PubMed query specifying randomized controlled studies and comparative studies on drug therapy.

Sentences containing direct comparisons of the pharmacological actions of two drugs expressed in the target structures (comp1 and comp2) were extracted starting from the latest retrieved citation and continuing until 300 sentences with comparative structures had been examined. These were annotated with the PubMed ID of the citation, names of two drugs (COMPARED_WITH predication), the scale on which they are compared (SCALE), and the relative position of the primary drug with respect to the secondary (SAME_AS, HIGHER_THAN, or LOWER_THAN).

The test sentences were processed using SemRep and evaluated against the annotated test set. We then computed recall and precision in several ways: overall for all comparative structures, for comp1 structures only, and for comp2 structures only. To understand how the overall identification of comparatives is influenced by the components of the construction, we also computed recall and precision separately for drug names, scale, and position on scale (SAME_AS, HIGHER_THAN and LOWER_THAN taken together). Recall measures the proportion of manually annotated categories that have been correctly identified automatically. Precision measures what proportion of the automatically annotated categories is correct.

In addition, the overall identification of comparative structures was evaluated using the F-measure [Rijsbergen, 1979], which combines recall and precision. The F-measure was computed using macro-averaging and micro-averaging. Macro-averaging was computed over each category first and then averaged over the three categories (drug names, scale, and position on scale). This approach gives equal weight to each category. In micro-averaging (which gives an equal weight to the performance on each sentence) recall and precision

were obtained by summing over all individual sentences. Because it is impossible to enumerate all entities and relations which are not drugs, scale, or position we did not use the classification error rate and other metrics that require computing of true negative values.

4 Results

Upon inspection of the SemRep processing results we noticed that the test set contained nine duplicates. In addition, four sentences were not processed for various technical reasons. We report the results for the remaining 287 sentences, which contain 288 comparative structures occurring in 168 MEDLINE citations. Seventy four citations contain 85 comp2 structures. The remaining 203 structures are comp1.

Correct identification of comparative structures of both types depends on two factors: 1) recognition of both drugs being compared, and 2) recognition of the presence of a comparative structure itself. In addition, correct identification of the comp2 structures depends on recognition of the scale on which the drugs are compared and the relative position of the drugs on the scale. Table 1 presents recall, precision, and F-score reflecting these factors.

Table 1. SemRep performance

Task	Recall	Precision	F-score
Overall	0.70	0.96	0.81
Drug extraction	0.69	0.96	0.81
Comp1	0.74	0.98	0.84
Comp2	0.62	0.92	0.74
Scale	0.62	1.00	0.77
Position on scale	0.62	0.98	0.76

We considered drug identification to be correct only if both drugs participating in the relationship were identified correctly. The recall results indicate that approximately 30% of the drugs and comparative structures of comp1, as well as 40% of comp2 structures, remain unrecognized; however, all components are identified with high precision. Macro-averaging over compared drug names, scale, and position on scale categories we achieve an F-score = 0.78. The micro-average score for 287 comparative sentences is 0.5.

5 Discussion

In examining SemRep errors, we determined that more than 60% of the false negatives (for both comp1 and comp2) were due to "empty heads" [Chodorow et al., 1985; Guthrie et al., 1990], in which the syntactic head of a noun phrase does not reflect semantic thrust. Such heads prevent SemRep from accurately determining the semantic type and group of the noun phrase. In our sample, expressions interpreted as empty heads include those referring to drug dosage and formulations, such as *extended release* (the latter often abbreviated as *XR*). Examples of missed interpretations are in sentences (28) and (29), where the empty heads are in bold. Ahlers et al. [Ahlers et al., 2007] discuss enhancements to SemRep for accommodating empty heads. These mechanisms are being incorporated into the processing for comparative structures.

(28) Oxybutynin **15 mg** was more effective than propiverine **20 mg** in reducing symptomatic and asymptomatic IDCs in ambulatory patients.

(29) Intravesical atropine was as effective as oxybutynin **immediate release** for increasing bladder capacity and it was probably better with less antimuscarinic side effects

False positives were due exclusively to word sense ambiguity. For example, in (30) *bid* (twice a day) was mapped to the concept "BID protein", which belongs to the semantic group Chemicals & Drugs. The most recent version of MetaMap, which will soon be called by comparative processing, exploits word sense disambiguation [Humphrey et al., 2006] and will likely resolve some of these errors.

(30) Retapamulin ointment 1% (**bid**) for 5 days was as effective as oral cephalexin (**bid**) for 10 days in treatment of patients with SID, and was well tolerated.

Although, in this paper, we tested the method on structures in which the compared terms belong to the semantic group Chemicals & Drugs, we can straightforwardly generalize the method by adding other semantic groups to the algorithm. For

example, if SemRep recognized the noun phrases in bold in (31) and (32) as belonging to the group Procedures, comparative processing could proceed as for Chemicals & Drugs.

(31) Comparison of **multi-slice spiral CT** and **magnetic resonance** imaging in evaluation of the un-resectability of blood vessels in pancreatic tumor.

(32) Dynamic **multi-slice spiral CT** is better than dynamic **magnetic resonance** to some extent in evaluating the un-resectability of peripancreatic blood vessels in pancreatic tumor.

The semantic predications returned by SemRep to represent comparative expressions can be considered a type of executable knowledge that supports reasoning. Since the arguments in these predications have been mapped to the UMLS, a structured knowledge source, they can be manipulated using that knowledge. It is also possible to compute the transitive closure of all SemRep output for a collection of texts to determine which drug was asserted in that collection to be the best with respect to some characteristic. This ability could be very useful in supporting question-answering applications.

As noted earlier, it is common in reporting on the results of randomized clinical trials and systematic reviews that a comp1 structure appears early in the discourse to announce the objectives of the study and that a comp2 structure often appears near the end to give the results. Another example of this phenomenon appears in (33) and (34) (from PMID 15943841).

(33) To compare the efficacy of famotidine and omeprazole in Japanese patients with non-erosive gastro-oesophageal reflux disease by a prospective randomized multicentre trial.

(34) Omeprazole is more effective than famotidine for the control of gastro-oesophageal reflux disease symptoms in H. pylori-negative patients.

We suggest one example of an application that can benefit from the information provided by the knowledge inherent in the semantic interpretation of comparative structures, and that is the interpretation of outcome statements in MEDLINE citations, as a method for supporting automatic access to the latest results from clinical trials research.

6 Conclusion

We expanded a symbolic semantic interpreter to identify comparative constructions in biomedical text. The method relies on underspecified syntactic analysis and domain knowledge from the UMLS. We identify two compared terms and scalar comparative structures in MEDLINE citations. Although we restricted the method to comparisons of drug therapies, the method can be easily generalized to other entities such as diagnostic and therapeutic procedures. The availability of this information in computable format can support the identification of outcome sentences in MEDLINE, which in turn supports translation of biomedical research into improvements in quality of patient care.

Acknowledgement This study was supported in part by the Intramural Research Programs of the National Institutes of Health, National Library of Medicine.

References

Ahlers C, Fiszman M, Demner-Fushman D, Lang F, Rindflesch TC. 2007. Extracting semantic predications from MEDLINE citations for pharmacogenomics. *Pacific Symposium on Biocomputing* 12:209-220.

Aronson AR. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. *Proc AMIA Symp*, 17-21.

Blaschke C, Andrade MA, Ouzounis C, and Valencia A. 1999. Automatic extraction of biological information from scientific text: protein-protein interactions. *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*. Morgan Kaufman Publishers, San Francisco, CA.

Christensen L, Haug PJ, and Fiszman M. 2002. MPLUS: A probabilistic medical language understanding system. *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain, Association for Computational Linguistics,* 29-36.

Chodorow MS, Byrd RI, and Heidom GE. 1985. Extracting Semantic Hierarchies from a Large On-

Line Dictionary. *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, 299-304.

Chun HW, Tsuruoka Y, Kim J-D, Shiba R, Nagata N, Hishiki T, and Tsujii J. 2006, Extraction of gene-disease relations from Medline using domain dictionaries and machine learning. *Pac Symp Biocomput*, 4-15.

Friedman C. 1989. A general computational treatment of the comparative. *Proc 27th Annual Meeting Assoc Comp Linguistics*, 161-168.

Friedman C, Alderson PO, Austin JH, Cimino JJ, and Johnson SB. 1994. A general natural-language text processor for clinical radiology. *J Am Med Inform Assoc*, 1(2):161-74.

Friedman C, Kra P, Yu H, Krauthammer M, and Rzhetsky A. 2001. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 Suppl 1:S74-S82.

Guthrie L, Slater BM, Wilks Y, Bruce R. 1990. Is there content in empty heads? *Proceedings of the 13th Conference on Computational Linguistics*, v3:138 – 143.

Hahn U, Romacker M, and Schulz S. 2002. MEDSYNDIKATE--a natural language system for the extraction of medical information from findings reports. *Int J Med Inf*, 67(1-3):63-74.

Huddleston R, and Pullum GK. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge, UK.

Humphrey SM, Rogers WJ, Kilicoglu H, Demner-Fushman D, Rindflesch TC. 2006. Word sense disambiguation by selecting the best semantic type based on Journal Descriptor Indexing: Preliminary experiment. *J Am Soc Inf SciTech* 57(1):96-113.

Humphreys BL, Lindberg DA, Schoolman HM, and Barnett OG. 1998. The Unified Medical Language System: An informatics research collaboration. *J Am Med Inform Assoc*, 5(1):1-11.

Jindal, Nitin and Bing Liu. 2006. Identifying comparative sentences in text documents. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*.

Johnson SB, Aguirre A, Peng P, and Cimino J. 1993. Interpreting natural language queries using the UMLS. Pro*c Annu Symp Comput Appl Med Care*, 294-8.

Leroy G, Chen H, and Martinez JD. 2003 A shallow parser based on closed-class words to capture relations in biomedical text. *J Biomed Inform*, 36(3):145-158.

Lussier YA, Borlawsky T, Rappaport D, Liu Y, and Friedman C. 2006 PhenoGO: assigning phenotypic context to Gene Ontology annotations with natural language processing. *Pac Symp Biocomput*, 64-75.

McCray AT, Srinivasan S, and Browne AC. 1994. Lexical methods for managing variation in biomedical terminologies. *Proc Annu Symp Comput Appl Med Care*, 235-9.

McCray AT, Burgun A, and Bodenreider O. 2001 Aggregating UMLS semantic types for reducing conceptual complexity. *Medinfo*, 10(Pt 1): 216-20.

Rayner M and Banks A. 1990. An implementable semantics for comparative constructions. *Computational Linguistics*, 16(2):86-112.

Rindflesch TC. 1995. Integrating natural language processing and biomedical domain knowledge for increased information retrieval effectiveness. *Proc 5th Annual Dual-use Technologies and Applications Conference*, 260-5.

Rindflesch TC and Fiszman M. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: Interpreting hypernymic propositions in biomedical text. *J Biomed Inform*, 36(6):462-77.

Rindflesch TC, Marcelo Fiszman , and Bisharah Libbus. 2005. Semantic interpretation for the biomedical research literature. *Medical informatics: Knowledge management and data mining in biomedicine*. Springer, New York, NY.

Rijsbergen V. 1979. *Information Retrieval*, Butterworth-Heinemann, Newton, MA.

Ryan K. 1981. Corepresentational grammar and parsing English comparatives. *Proc 19th Annual Meeting Assoc Comp Linguistics*, 13-18.

Smith L, Rindflesch T, and Wilbur WJ. 2004. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320-1.

Staab S and Hahn U. Comparatives in context. 1997. *Proc 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference*, 616-621.

Yen YT, Chen B, Chiu HW, Lee YC, Li YC, and Hsu CY. 2006. Developing an NLP and IR-based algorithm for analyzing gene-disease relationships.

The Extraction of Enriched Protein-Protein Interactions from Biomedical Text

Barry Haddow and Michael Matthews
School of Informatics
University of Edinburgh
2 Buccleuch Place
Edinburgh, Scotland, EH8 9LW
{bhaddow,mmatsews}@inf.ed.ac.uk

Abstract

There has been much recent interest in the extraction of PPIs (protein-protein interactions) from biomedical texts, but in order to assist with curation efforts, the PPIs must be enriched with further information of biological interest. This paper describes the implementation of a system to extract and enrich PPIs, developed and tested using an annotated corpus of biomedical texts, and employing both machine-learning and rule-based techniques.

1 Introduction

The huge volume of literature generated in the biomedical field is such that researchers are unable to read all the papers that interest them. Instead they must rely on curated databases, containing information extracted from the literature about, for example, which proteins interact.

These curated databases are expensive to produce as they rely on qualified biologists to select the papers, read them to extract the relevant information, enter this information into the database, and cross-check the information for quality control, a procedure which can be very time-consuming. If NLP techniques could be used to aid curators in their task then the costs of producing curated databases could be substantially reduced.

In the context of biomedical information extraction, there has been much recent interest in the automated extraction of PPIs (protein-protein interactions) from biomedical literature. The recent BioCreAtIvE Challenge highlights the desire to utilize these extraction techniques to automatically or semi-automatically populate curated PPI databases. However, just identifying the interactions is not necessarily sufficient, as curators typically require additional information about the interactions, such as the experimental method used to detect the interaction, and the names of any drugs used to influence the behaviour of the proteins. Furthermore, curators may only be interested in interactions which are experimentally proven within the paper, or where the proteins physically touch during the interaction.

This paper describes the implementation of a system designed to extract mentions of PPIs from biomedical text, and to enrich those PPIs with additional information of biological interest. The enriched information consists of properties (name-value pairs associated with a PPI, for example a directness property could indicate whether the interaction is direct or not direct) and attributes (relations between the PPI relation or its participating entities and other entities, such as the experimental method used to detect the PPI). This system for extracting and enriching PPIs was developed as part of the TXM programme, which aims to develop tools to help with the curation of biomedical papers.

After reviewing related work in the following section, a detailed description of how the annotated corpus was created and its descriptive statistics is provided in section 3. The methods used to extract the properties and attributes are explained in section 4, and then evaluated and discussed in section 5. Some conclusions and suggestions for further work are offered in section 6.

BioNLP 2007: Biological, translational, and clinical language processing, pages 145–152,
Prague, June 2007.

2 Related Work

There has been much recent interest in extracting PPIs from abstracts and full text papers (Bunescu and Mooney, 2006; Giuliano et al., 2006; Plake et al., 2005; Blaschke and Valencia, 2002; Donaldson et al., 2003). In these systems however, the focus has been on extracting just the PPIs without attempts to enrich the PPIs with further information. Enriched PPIs can be seen as a type of biological event extraction (Alphonse et al., 2004; Wattarujeekrit et al., 2004), a technique for mapping entities found in text to roles in predefined templates which was made popular in the MUC tasks (Marsh and Perzanowski, 1998). There has also been work to enrich sentences with semantic categories (Shah and Bork, 2006) and qualitative dimensions such as polarity (Wilbur et al., 2006).

Using NLP to aid in curation was addressed in the KDD 2002 Cup (Yeh et al., 2002), where participants attempted to extract records curatable with respect to the FlyBase database, and has been further studied by many groups (Xu et al., 2006; Karamanis et al., 2007; Ursing et al., 2001).

The Protein-Protein Interaction task of the recent BioCreAtIvE challenge (Krallinger et al., 2007) was concerned with selecting papers and extracting information suitable for curation. The PPI detection subtask (IPS) required participants not simply to detect PPI mentions, but to detect curatable PPI mentions, in other words to enrich the PPI mentions with extra information. Furthermore, another of the subtasks (IMS) required participants to add information about experimental methods to the curatable PPIs.

3 Data Collection and Corpus

3.1 Annotation of the Corpus

A total of 217 papers were selected for annotation from PubMed and PubMedCentral as having experimentally proven protein-protein interactions (PPIs). The papers were annotated by a team of nine annotators, all qualified in biology to at least PhD level, over a period of approximately five months.

The XML versions of the papers were used wherever possible, otherwise the HTML versions were used and converted to XML using an in-house tool. The full-text of each paper, including figure captions, was annotated, although the materials and methods sections were not included in the annotation.

From the 217 annotated papers, a total of 65 were selected randomly for double annotation and 27 for triple annotation. These multiply-annotated papers were used to measure inter-annotator agreement (IAA), by taking each pair of annotations on the same paper, and scoring one annotation against the other using the same algorithm as for scoring the system against the annotated data (see Section 5). Each doubly annotated paper contributed one pair of annotations, whilst the triply annotated papers contributed three pairs of annotations. The overall IAA score is the micro-average of the F_1 scores on each pair of corresponding annotations, where it should be emphasised that the F_1 does not depend on the order in which the annotated papers were combined. The multiply annotated papers were not reconciled to produce a single gold version, rather the multiple versions were left in the corpus.

The papers were annotated for entities and relations, and the relations were enriched with properties and attributes. The entities chosen for annotation were those involved in PPIs (Protein, Complex, Fusion, Mutant and Fragment) and those which could be attributes of PPIs (CellLine, DrugCompound, ExperimentalMethod and ModificationType). A description of the properties and attributes, as well as counts and IAA scores are shown in Tables 1 and 2.

Once annotated, the corpus was split randomly into three sections, TRAIN (66%), DEVTEST (17%) and TEST (17%). TRAIN and DEVTEST were to be used during the development of the system, for feature exploration, parameter tuning etc., whilst TEST was reserved for scoring the final system. The splits were organised so that multiply annotated versions of the same paper were placed into the same section.

3.2 Descriptive Statistics of Corpus

The total number of distinct PPIs annotated in the 336 papers was 11523, and the PPI IAA, measured using F_1, was 64.77. The following are examples of enriched PPIs, with the entities in bold face:

(1) **Tat** may also increase initiation of HIV-1 transcription by enhancing **phosphorylation** of **SP1**, a transcription factor involved in the basal HIV-1 transcription [14].

Name	Explanation	Values	Counts	Pct	IAA
IsPositive	The polarity of the statement about the PPI.	Positive	10718	93.01	99.57
		Negative	836	7.26	90.12
IsDirect	Whether the PPI is direct or not.	Direct	7599	65.95	86.59
		NotDirect	3977	34.51	61.38
IsProven	Whether the PPI is proven in the paper or not.	Proven	7562	65.63	87.75
		Referenced	2894	25.11	88.61
		Unspecified	1096	9.51	34.38

Table 1: The properties that were attached to PPIs, their possible values, counts and IAA

Name	Entity type	Explanation	Count	IAA
InteractionDetectionMethod	ExperimentalMethod	Method used to detect the PPI.	2085	59.96
ParticipantIdentificationMethod	ExperimentalMethod	Method used to detect the participant.	1250	36.83
ModificationBefore	Modification	Modification of participant before interaction.	240	68.13
ModificationAfter	Modification	Modification of participant after interaction.	1198	86.47
DrugTreatment	DrugCompound	Treatment applied to participant.	844	49.00
CellLine	CellLine	Cell-line from which participant was drawn.	2000	64.38

Table 2: The attributes that could be attached to the PPIs, with their entity type, counts and IAA

(2) To confirm that **LIS1** and **Tat** interact in vivo, we used **yeast two-hybrid system**, in which **Tat** was expressed as a bait and **LIS1** as a prey. Again, we found that **LIS1** and **Tat** interacted in this system.

In Example 1, the properties attached to the PPI between "Tat" and "SP1" are Referenced, Direct and Positive, and "phosphorylated" is attached as a ModificationAfter attribute. Example 2 shows a PPI between "Tat" and "LIS1" (in the second sentence) which is given the properties Proven, Direct and Positive, and has the InteractionDetectionMethod attribute "yeast two-hybrid system". This second example indicates that attributes do not have to occur in the same sentence.

Statistics on the occurrence of properties are shown in Table 1. For most of the property values, there are significant numbers of PPIs, except for Unspecified and Negative, which are used in less than 10% of cases. Note that annotators were permitted to mark more than one PPI between a given pair of entities if, for example, they wished to mark both Positive and Negative PPIs because the author is making a statement that proteins interact under one condition and not under another condition. For the purposes of data analysis and to make modelling easier, such PPIs have been collapsed to give a single PPI which may have multiple values for each property and attribute.

Table 2 shows occurrence statistics for attributes, where, as for properties, there can be multiple values for the same attribute. A notable feature of the attribute attachment counts is that certain attributes (ModificationBefore and DrugTreatment especially) are quite rarely attached, making it difficult to use statistical techniques.

Also shown in Tables 1 and 2 are the IAA figures for all properties and attributes. The IAA for properties is generally high, excepted for the Unspecified value of the IsProven property. This being something of a "none of the above" category means that the annotators probably have different standards re-

garding the uncertainty required before the PPI is placed in this class. The IAA for attributes is, on the whole, lower, with some attributes showing particularly low IAA (ParticipantIdentificationMethod). A closer investigation shows that the bulk of the disagreement is about when to attach, in other words if both annotators decide to attach an attribute to a particular PPI, they generally agree about which one, scoring a micro-averaged overall F_1 of 95.10 in this case.

4 Methods

4.1 Pipeline Processing

The property and attribute assignment modules were implemented as part of an NLP pipeline based on the LT-XML2 architecture[1]. The pipeline consists of tokenisation, lemmatisation, part-of-speech tagging, species word identification, abbreviation detection and chunking, named entiry recognition (NER) and relation extraction. The part-of-speech tagging uses the Curran and Clark POS tagger (Curran and Clark, 2003) trained on MedPost data (Smith et al., 2004), whilst the other preprocessing stages are all rule based. Tokenisation, species word identification and chunking were implemented in-house using the LT-XML2 tools (Grover and Tobin, 2006), whilst abbreviation extraction used the Schwartz and Hearst abbreviation extractor (Schwartz and Hearst, 2003) and lemmatisation used morpha (Minnen et al., 2000).

The NER module uses the Curran and Clark NER tagger (Curran and Clark, 2003), augmented with extra features tailored to the biomedical domain. Finally, a relation extractor based on a maximum entropy model and a set of shallow linguistic features is employed, as described in (Nielsen, 2006).

4.2 Properties

To assign properties to each PPI extracted by the relation extraction component, a machine learning based property tagger was trained on a set of features extracted from the context of the PPI. The property tagger used a separate classifier for each property, but with the same feature set, and both Maximum Entropy (implemented using Zhang Le's maxent[2]) and Support Vector Machines (implemented using svmlight[3]) were tested. To choose an optimal feature set, an iterative greedy optimisation procedure was employed. A set of potential features were implemented, with options to turn parts of the feature set on or off. The full feature set was then tested on the DEVTEST data with each of the feature options knocked out in turn. After examining the scores on all possible feature knockouts, the one which offered the largest gain in performance was selected and removed permanently. The whole procedure was then repeated until knockouts produced no further gains in performance. The resulting optimised feature set contains the following features:

ngram Both unigrams and bigrams were implemented, although, after optimisation, unigrams were switched off. The ngram feature uses *vlw backoff*, which means that words are replaced by their verb stems, backed off to lemmas and then to the word itself if not available. Furthermore, all digits in the words are replaced with "0". Ngrams are extracted from the sentences containing the participants in the PPI, and all sentences in between. Ngrams occurring before, between and after the participants of the PPI are treated as separate features.

entity The entity feature includes the text and type of the entities in the PPI.

headword This feature is essentially constructed in the same way as the ngram feature, except that only head verbs of chunks in the context are included, and the vlw backoff is not used.

entity-context In the entity context feature, the vlw backoffs of the two words on either side of each of the entities in the PPI are included, with their positions marked.

4.3 Attributes

For attribute assignment, experiments were performed with both rule-based and machine-learning approaches. The following sections summarise the methods used for each approach.

4.3.1 Rule-based

In the rule-based approach, hand-written rules were written for each attribute, using part-of-speech tags, lemmas, chunk tags, head words and the NER tags. In all, 20 rules were written. Each rule is

[1] http://www.ltg.ed.ac.uk/software/xml/

[2] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

[3] http://svmlight.joachims.org/

Rule	Protein	Prec	Count
P1 ATT P2	P2	100	13
P1 is *ATT* by *P2*	P1	100	1
ATT of *P2*	P2	86.1	112
ATT of *P1*	P1	74.5	80
*P1 * ATT* site	P1	72.2	13
*P1 * ATT* by * *P2*	P2	70.0	100
*P1 * (ATT* pass) * *P1*	P2	64.0	16
*P1 * ATT * P2*	P2	67.5	187
P2 ATT	P2	75.0	100
P2 - any-word ATT	P1	73.7	14

Table 3: The rules used to assign ModificationAfter attributes. The protein column indicates whether the attribute attaches to the 1st or 2nd protein, the prec field indicates the precision of the rule on the training set and the count indicates the number of times the rule applied correctly in training. In the rules, **P1** refers to the first protein, **P2** refers to the second protein, **ATT** refers to the attribute, * refers to any number of words, *any-word* refers to any single word, and pass refers to the passive voice. For example, the rule "*P2 - any-word ATT*" applied to the sentence "protein 1 is regulated by protein 2-dependent phosphorylation" would result in the attribute *phosphorylation* being assigned as the ModificationAfter attribute to *protein 1*.

ranked according to its precision as determined on the TRAIN set, and the rules are applied in order of their precision. This is particularly important with modification attributes which are constrained so that a given modification entity can only be attached once per interaction. Table 3 lists the rules used to assign the ModificationAfter attribute.

4.3.2 Machine Learning

For this approach, attributes are modelled as relations between PPIs and other entities. For each PPI in a document, a set of candidate relations is created between each of the entities in the PPI and each of the attribute entities contained in the same sentence(s) as the PPI[4]. If there are no entities of the appropriate type for a given attribute in the same sentence as the PPI, the sentences before and after the PPI are also scanned for candidate entities. Each of the candidate relations that correspond to attributes annotated in the gold standard are considered positive examples, whilst those that were not annotated are considered negative examples. For example, given the following sentence:

```
Protein A phosphorylates protein B
[Protein] [Modification] [Protein]
```

If the gold standard indicates a PPI between Protein A and Protein B with phosphorylates assigned as a ModificationAfter attribute to Protein B, four candidate relations will be created as shown in Table 4

Type	Entity 1	Entity 2	Label
Mod Before	Prot A	phosphorylates	neg
Mod Before	Prot B	phosphorylates	neg
Mod After	Prot A	phosphorylates	neg
Mod After	Prot B	phosphorylates	pos

Table 4: Candidate Attribute Relations for Protein A phosphorylates Protein B

A set of features is extracted for each of the examples and a maximum entropy (ME) model is trained using Zhang Le's maxent toolkit. The features used are listed below:

entity The text and part-of-speech of the attribute, as used for properties.

entity-context The entity context feature used for properties, except that the context size was increased to 4, and parts-of-speech of the context words were also included.

ngram This is the same as the ngram feature used for properties, except that unigrams were switched on.

entities-between The entities that appear between the two entities involved in the candidate relation.

parent-relation-feature Indicates the position of the attribute entity with respect to parent PPI (i.e. before, after, or in between). For attributes that are in between the two entities involved in the PPI, also indicates if the sentence is active or passive.

5 Evaluation

5.1 Properties

To score the property tagger, precision, recall and F_1 are calculated for each of the seven possible

[4] PPIs spanning more than 2 sentences were ignored

Name	Value	Baseline		Maximum Entropy		SVM	
		Gold	Predicted	Gold	Predicted	Gold	Predicted
IsPositive	Positive	96.87	97.33	97.10	98.22	97.08	98.27
	Negative	0.00	0.00	38.46	48.39	45.45	57.53
IsDirect	Direct	78.66	81.90	82.05	85.54	81.94	86.87
	NotDirect	0.00	0.00	58.92	54.33	60.80	63.44
IsProven	Proven	78.21	78.85	87.86	82.73	88.08	88.51
	Referenced	0.00	0.00	81.46	69.65	82.83	81.97
	Unspecified	0.00	0.00	25.74	29.41	22.77	28.00
Overall		74.20	76.24	83.87	83.33	84.09	86.79

Table 5: The performance of the property tagger, measured by training on TRAIN and DEVTEST combined, then testing on TEST. The two scores given for each system are for testing on gold PPIs, and testing on predicted PPIs. An F_1 score is shown for each property value, as well as a microaveraged overall score.

property values and then the F_1 scores are microaveraged to give an overall score. As mentioned in Section 3.1, all versions of the annotation for each multiply-annotated document were included in the training and test sets, taking care that all versions of the same document were included in the same set. This has the disadvantage that the system can never achieve 100% in cases where the annotators differ, but the advantage of giving partial credit where there is genuine ambiguity and the system agrees with one of the options chosen by the annotators.

The scores for all property values, tested on TEST, are shown in Table 5, both using the model (with Maximum Entropy and SVM) and using a baseline where the most popular value is assigned. Two scores are shown, the performance as measured when the test set has the gold PPIs, and the performance when the test set has the predicted PPIs, scored only on those PPIs where both system and gold agree. The relation extractor used to predict the PPIs is trained on the same documents as were used to train the property tagger.

To see which features were most effective, a knockout (lesion) test was conducted in which features were knocked out one by one and performance was measured on the DEVTEST set. In each feature knockout, one of the features from the list in Section 4.2 was removed. Table 6 shows how the overall performance is affected by the different knockouts. From the knockout experiment it is clear that the ngram (actually bigram) feature is by far the most effective, with the other features only contributing marginally to the results.

Feature	Knockout score	Difference
vanilla	86.08	0.00
ngram	81.86	-4.22
entity	85.30	-0.77
headword	84.38	-0.50
entity-context	85.54	-0.54

Table 6: The effect of knocking out features on the property score. Tests are conducted by training on TRAIN and testing on DEVTEST, on predicted PPIs. "vanilla" refers to the case where the optimal features set is employed.

5.2 Attributes

The attributes are scored in the same manner as the properties. Table 7 summarises the results for both the rule-based and machine learning attribute systems. These are compared to a baseline system that simply attaches the nearest entity of the appropriate type for each attribute.

5.3 Discussion

The results for the more common property values are generally close to human performance (as measured by IAA), however performance on both IsNegative and Unspecified is fairly low. In the case of Unspecified, the IAA is also low, making it likely that the training and test data is inconsistent, compounding the problem of the low occurrence rate of this value. The Negative value also suffers from a low occurrence rate, leading to an imbalance between Negative and Positive which makes life hard for the

Attribute	Baseline		Rule-based		Machine Learning	
	Gold	Predicted	Gold	Predicted	Gold	Predicted
InteractionDetectionMethod	36.02	39.71	39.22	41.38	37.02	46.81
ParticipantIdentificationMethod	08.68	09.27	12.32	12.87	03.37	05.97
ModificationBefore	13.10	16.00	42.22	43.84	04.88	08.33
ModificationAfter	43.37	46.00	64.93	73.04	62.32	69.64
DrugTreatment	49.57	51.11	51.29	53.33	13.90	24.52
CellLine	50.19	45.90	54.47	50.47	45.13	42.28
Overall	29.68	30.32	45.26	48.32	32.08	43.11

Table 7: The performance of the attribute tagger, on TEST. The two scores given for each system are for testing on gold PPIs, and testing on predicted PPIs. Performance on each attribute value is measured using F_1, and then microaveraged to give an overall figure.

machine learners. However it is also possible that the shallow linguistic features used in these experiments are not sufficient to make the sometimes subtle distinction between a negative statement about an interaction and a positive one, and that models based on a deeper linguistic analysis (e.g. parse trees as in (Moschitti, 2004)) would be more successful. Note also that the feature set was optimised for maximum performance across all property values, with all given equal weight, but if some values are more important than others then this could be taken into account in the optimisation, with possibly different feature sets used for different property names.

The results for the attributes using the rule-based system are approximately 75% of human performance and are higher than results for the machine learning system. However, for the ModificationAfter, CellLine, and InteractionDetectionMethod attributes, which occur more frequently than the other attributes and have higher IAA, the machine learning system is competitive and even slightly outperforms in the case of the InteractionDetectionMethod. The scores are directly correlated with the IAA and both the scores and the IAA are higher for the attributes that tend to occur in the same sentence as the PPI. On a practical level, this suggests that those who hope to create similar systems would be advised to start with local attributes and pay particular attention to IAA on non-local attributes.

5.4 Further work

As regards properties, good results were obtained using shallow linguistic features, but it would be interesting to learn whether machine learning techniques based on a deeper linguistic analysis would be more effective. Also, properties were treated as additional information added on to the PPIs after the relation extractor had run, but perhaps it would be more effective to combine relation extraction and property tagging to, for example, consider positive and negative PPIs as different types of relations.

For attributes, it would be interesting to combine the rule-based and machine learning systems. This has the advantage of having a system that can both learn from annotated data when it exists, but can be potentially improved by rules when necessary or when annotated data is not available. Another issue may be that some attributes might not be represented explicitly by a single entity in a document. For example, an experimental method may be described rather than explicitly stated. Attributes that are not local to the PPI caused difficulty for both the annotators and the system. It would be interesting to see if it is easier to attach attributes to a single PPI that has been derived from the text, rather than attempting to assign attributes to each specific mention of a PPI within the text. This could be accomplished by attempting to merge the information gathered from each relation along the lines described in (Hobbs, 2002)

Since the main motivation for developing the system to extract enriched PPIs was to develop a tool to aid curators, it would be useful to know how effective the system is in this task. Aside from (Karamanis et al., 2007), there has been little work published to date on the effect that NLP could have on the curation process. In the most recent BioCreAtIvE evaluation, the PPI subtasks were concerned with au-

tomating information extraction tasks typically performed by curators such as distinguishing between curatable and non-curatable PPI mentions and specifying the details of how the PPI was detected.

6 Conclusions

A system was implemented for enriching protein-protein interactions (PPIs) with properties and attributes providing additional information useful to biologists. It was found that a machine learning approach to property tagging, using simple contextual features, was very effective in most cases, but less effective for values that occurred rarely, or for which annotators found difficulty in assigning values. For the attributes, sparsity of data meant that rule-based approaches worked best, using fairly simple rules that could be quickly developed, although machine learning approaches could be competitive when there was sufficient data.

7 Acknowledgements

The authors are very grateful to the annotation team, and to Cognia (`http://www.cognia.com`) for their collaboration on the TXM project. This work is supported by the Text Mining Programme of ITI Life Sciences Scotland (`http://www.itilifesciences.com`).

References

Erick Alphonse, Sophie Aubin, Philippe Bessieres, Gilles Bisson, Thierry Hamon, Sandrine Lagarrigue, Adeline Nazarenko, Alain-Pierre Manine, Claire Nedellec, Mohamed Ould Abdel Vetah, Thierry Poibeau, and Davy Weissenbacher. 2004. Event-based information extraction for the biomedical domain: the Caderige project.

C. Blaschke and A. Valencia. 2002. The frame-based module of the suiseki information extraction system. *IEEE Intelligent Systems*, (17):14–20.

Razvan Bunescu and Raymond Mooney. 2006. Subsequence kernels for relation extraction. In Y. Weiss, B. Schlkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. Cambridge, MA.

James R. Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-2003*.

Ian Donaldson, Joel Martin, Berry de Bruijn, Cheryl Wolting, Vicki Lay, Brigitte Tuekam, Shudong Zhang, Berivan Baskin, Gary D. Bader, Katerina Michalickova, Tony Pawson, and Christopher W. V. Hogue. 2003. PreBIND and Textomy - mining the biomedical literature for protein-protein interactions using a support vector machine. *BMC Bioinformatics*, 4:11.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the EACL*.

Claire Grover and Richard Tobin. 2006. Rule-Based Chunking and Reusability. In *Proceedings of LREC 2006*.

Jerry R. Hobbs. 2002. Information extraction from biomedical text. *Journal of Biomedical Informatics*, 35(4):260–264.

N. Karamanis, I. Lewin, R. Seal, R. Drysdale, and E. J. Briscoe. 2007. Integrating natural language processing with flybase curation. In *Proceedings of PSB 2007*.

Martin Krallinger, Florian Leitner, and Alfonso Valencia. 2007. Assessment of the Second BioCreative PPI Task: Automatic Extraction of Protein-Protein Interactions. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

E. Marsh and D. Perzanowski. 1998. MUC-7 evaluation of IE technology: Overview of results. In *Proceedings of MUC-7*.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of INLG 2000*.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the ACL*.

Leif Arda Nielsen. 2006. Extracting protein-protein interactions using simple contextual features. In *Proceedings of the BioNLP 2006 at HLT/NAACL 2006.*

Conrad Plake, Jörg Hakenberg, and Ulf Leser. 2005. Optimizing syntax-patterns for discovering protein-protein-interactions. In *Proc ACM Symposium on Applied Computing, SAC, Bioinformatics Track*, volume 1, March.

A.S. Schwartz and M.A. Hearst. 2003. Identifying abbreviation definitions in biomedical text. In *Proceedings of PSB 2003.*

Parantu K. Shah and Peer Bork. 2006. Lsat: learning about alternative transcripts in medline. *Bioinformatics*, 22(7):857–865.

L. Smith, T. Rindflesch, and W. J. Wilbur. 2004. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321.

Björn M. Ursing, Frank H. J. van Enckevort, Jack A. M. Leunissen, and Roland J. Siezen. 2001. Exprot - a database for experimentally verified protein functions. *In Silico Biology*, 2:1.

Tuangthong Wattarujeekrit, Parantu K. Shah, and Nigel Collier. 2004. PASBio: predicate-argument structures for event extraction in molecular biology. *BMC Bioinformatics*, 5:155.

John W. Wilbur, Andrey Rzhetsky, and Hagit Shatkay. 2006. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC Bioinformatics*, 7:356+, July.

H. Xu, D. Krupke, J. Blake, and C. Friedman. 2006. A natural language processing (nlp) tool to assist in the curation of the laboratory mouse tumor biology database. *AMIA Annu Symp Proc*.

Alexander Yeh, Lynette Hirschman, and Alexander Morgan. 2002. Background and overview for KDD cup 2002 task 1: information extraction from biomedical articles. *SIGKDD Explor. Newsl.*, 4(2):87–89.

What's in a gene name? Automated refinement of gene name dictionaries

Jörg Hakenberg
Bioinformatics Group, Biotechnological Centre
Technische Universität Dresden, 01307 Dresden, Germany
hakenberg@informatik.hu-berlin.de

Abstract

Many approaches for named entity recognition rely on dictionaries gathered from curated databases (such as Entrez Gene for gene names.) Strategies for matching entries in a dictionary against arbitrary text use either inexact string matching that allows for known deviations, dictionaries enriched according to some observed rules, or a combination of both. Such refined dictionaries cover potential structural, lexical, orthographical, or morphological variations. In this paper, we present an approach to automatically analyze dictionaries to discover how names are composed and which variations typically occur. This knowledge can be constructed by looking at single entries (names and synonyms for one gene), and then be transferred to entries that show similar patterns in one or more synonyms. For instance, knowledge about words that are frequently missing in (or added to) a name ("antigen", "protein", "human") could automatically be extracted from dictionaries. This paper should be seen as a vision paper, though we implemented most of the ideas presented and show results for the task of gene name recognition. The automatically extracted name composition rules can easily be included in existing approaches, and provide valuable insights into the biomedical sub-language.

1 Introduction

Recognition of named entities (NER), such as names referring to genes and proteins, forms a major building block for text mining systems. Especially in the life sciences, a large amount of different entity types and their instances exist. Two basic strategies for NER are classification- and dictionary-based approaches. Classifiers learn (or are given) models to decide whether a sequence of tokens refers to an entity or not. Such decisions are based on various forms of input, for instance, tokens and their sequence in a sentence, part-of-speech tags, characteristic suffixes, and trigger keywords[1] (Hakenberg et al., 2005). Models can be learned from a given training sample. Dictionary-based approaches rely on curated word lists containing (all known) representatives of an entity type. Manual or automated refinement of the dictionary and inexact matching strategies allow to cover a broad spectrum of name variations (Hanisch et al., 2005). Classification-based approaches have proven to be very robust towards unseen tokens and names, because they also incorporate knowledge on names of the given class in general[1] (Crim et al., 2005). Dictionaries, on the other hand, reflect the knowledge about an entity class at a given time, and such approaches cannot find instances unknown to them. However, the main advantage of dictionary-based NER is that they bring the explicit possibility to map recognized entities to the source of the entries (most times, a database.) This alleviates the task of named entity

[1] For example, a protein name often is/has a proper noun; many enzymes end with '–ase'; 'domain of' is often followed by a protein name.

BioNLP 2007: Biological, translational, and clinical language processing, pages 153–160,
Prague, June 2007.

identification (NEI) that is needed to annotate texts properly or link text-mined facts to database entries.

In this paper, we want to concentrate on dictionary-based approaches and present ideas of how these could be automatically refined and enriched. In such a setting, named entity recognition functions as a method of 'spotting' entities in a text, after which further identification (disambiguation) is needed. NER components thus should guarantee very high recall rates with a reasonable precision. NEI then refines the predictions of NER, eliminating false positive annotations and identifying names. That such a setup would perform quite well is reflected, for example, in a study presented by Xu et al. (2007). They showed that sophisticated disambiguation strategies currently yield up to 93.9% precision (for mouse genes; yeast: 89.5%, fly: 77.8%.) Participants in the BioCreAtIvE 2 challenge showed similar values for human genes (up to 84.1% precision, 87.5% recall, or 81.1% F1), see Morgan and Hirschman (2007) for a summary.

Hand-coded rules for creating spelling variations have been proposed before, see section on Related Work. Such rules are applied to synonyms to generate morphological and orthographical variations ("Fas ligand" → "Fas ligands" and "Ifn gamma" → "Ifn-γ", respectively). In the same manner, systems use known patterns for structural changes of names and mappings for lexical variations to enrich existing dictionaries ("CD95R" → "receptor of CD95" and "gastric alcohol dehydrogenase" → "stomach alcohol dehydrogenase"). Our research question in this paper is, how such rules can be learned automatically from dictionaries that contain entries of the same entity class with multiple, typical synonyms each. Learning about the composition of names comes down to an analysis of known names. A human, given the same task, would look through a lot of examples to derive term formation patterns. Questions to ask are:

- What are frequent orthographical and morphological variations?
- Which parts of a name get abbreviated?
- How are abbreviations formed?
- Which identical abbreviations can be observed in multiple names?
- In which way can a name structurally and lexically change?
- Which are the parts of a name that can be exchanged with other terms or skipped entirely?
- Which are the important parts of a name, which are additional descriptive elements?

In this paper, we demonstrate methods to analyze names in order to find the semantically important parts. We map these parts to potential syntactic variations thereof observed within a name and its synonyms. We assess the frequency of such mappings (exchange of tokens, different ordering of tokens, etc.) and transfer this knowledge to all other names in the same dictionary. In this setup, understanding a name results in a structured decomposition of the name. Such decompositions provide knowledge on how to find (and identify) the name in arbitrary text, as they give insights into its mandatory, unique, and ambiguous[2] parts.

This paper should be seen as a vision paper, though we implemented most of the ideas presented herein and show first results. We first explain the idea behind learning name composition rules, motivated by manual curation as described in Related Work. We then explain the basic techniques needed for our analysis. We show how single entries (a name and all its synonyms) can be analyzed to find composition rules, and how these can be transferred to other entries. Preliminary results using some of the ideas presented here are also given. We conclude this paper with a discussion of the experimental methodology and an outlook.

1.1 Related Work

Current survey articles cover the spectrum of recent methods and results for biomedical named entity recognition and identification (Cohen and Hersh, 2005; Leser and Hakenberg, 2005). A recent assessment of named entity recognition and identification was done during the BioCreAtIvE 2 evaluation[3]. Official results will be available in April 2007. Naturally, a number of systems proposed before are highly related to the method presented in this paper. Hanisch et al. (2005) proposed the ProMiner system to recognize and identify protein names in text. They observed that the ordering of tokens in a name occur quite frequently, but do not change the seman-

[2] The latter two as compared to the whole dictionary.

[3] See http://biocreative.sourceforge.net .

tics of the overall name. They presented a model for protein names, partitioning tokens into token classes according to their semantic significance: modifiers ("receptor"), specifiers ("alpha"), non-descriptive tokens ("fragment"), standard tokens ("TNF"), plus common English words and interpunctuation. To evaluate the significance of tokens, they count their respective frequencies in a dictionary. Hanisch et al. extract a dictionary using various knowledge source (HGNC etc.) and expand and prune it afterwards. Expansion and pruning are based on manually defined rules (separating numbers and words, expanding known unambiguous synonyms with known synonyms, applying curation lists maintained by biological experts, predefined regular expressions). The final matching procedure found names by comparing (expanded) tokens and their classes to arbitrary text, where some token classes were mandatory for the identification and others could be missing. ProMiner yielded results between 78 and 90% F1-measure on the BioCreAtIvE 1 (Task 1B), depending on the organism-specific sub-task. The highest recall was found to be 84.1% for fly, 81.4% for mouse, and 84.8% for yeast genes.

We used a similar method, relying entirely on manually defined rules for name variations, for the BioCreAtIvE 2 GN task (Hakenberg et al., 2007). We expanded the dictionary applying these rules to every synonym (treating abbreviations and spelled-out names slightly different). This yielded a recall of 92.7 and 87.5% on the training and test sets, respectively (F1: 81.1%). In the aftermath of BioCreAtIvE 2, we now try to improve this high recall values further, by automatically analyzing the whole dictionary of gene names instead of manually composing useful rules in a trial–and–error approach.

2 Methods

We first want to present the overall idea of learning name composition rules, guided by specific examples. We first show how comparison of synonyms known for one gene name yields insights into the 'meaning' of the gene, and produces rules for structural and lexical variations of its name(s). Afterwards, we explain how such rules can be exchanged between different genes and add to the understanding of each genes 'meaning.'

2.1 Techniques

We apply several techniques to the analysis of names. To detect abbreviations by pairwise comparison of synonyms, we use the algorithm proposed by Schwartz and Hearst (2003) as the core component[4]. We changed some of the details so that, for instance, the first letter of the potential abbreviation has to match the first letter of the proposed long form. We perform the detection of abbreviations not only on whole synonyms, but also on parts of each name (like for "TNF-alpha stimulated ABC protein"), so that this property of Schwartz and Hearst's algorithm (S&H) is recovered. A trivial adaptation also reveals which parts of an abbreviation (one or more characters) map to which parts of the long form (one token, one partial token.) As S&H allows for missing tokens in the long form, we can also add the possibility for (few) characters in the abbreviation not being reflected in the long form.

To detect inexact matches (that is, slight variations in morphology or orthography), we use an adaptation of the biological sequence alignment algorithm (Needleman and Wunsch, 1970). Using the computed alignment score, this yields an immediate quantification of the similarity of two terms.

We compare the sequence of identified name parts (parts of a name where a mapping from this part to a part of the other synonym exists) in order to find parts that can be skipped or exchanged with each other. In addition, this yields insights into potential permuations of all parts of a name, and shows where certain parts typically do or do not occur.

2.2 Representation

Representation of information extracted by parsing *i)* a synonym or *ii)* all synonyms of a gene becomes a crucial basic part of our approach. *Concepts* have to be found in a name, for instance,

- *substance*: "serotonin",
- *type*: "receptor",
- *function*: "transcription factor", or
- *family-member*: "family-member number 6".

Also, for these concepts, rules have to be learned that match them against text (or vice versa): an 'R' hints on a receptor, a '6' at the end of a name (for instance, a noun phrase) hints on a family-member or

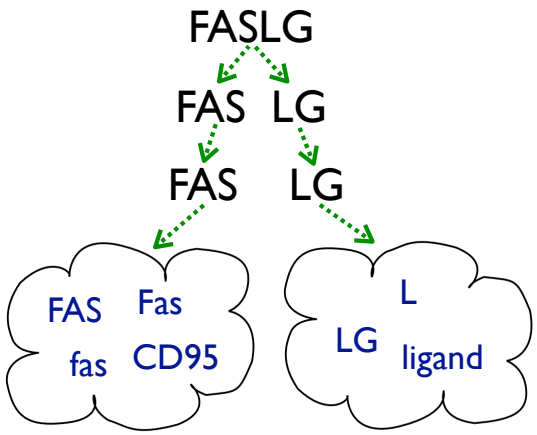
[4]The original algorithm decides whether a given short form can be explained by a given long form.

Type	Example token	Example name
Descriptor	antigen, ligand, inhibitor	P-30 antigen
Modifier	factor, family member, type	BRG1-associated factor
Specifier	alpha, IX, A	TNF alpha
Source	d, _HUMAN, p	dHNF-4

Table 1: Types of tokens that frequently occur in gene names. Also see Hanisch et al. (2005), though they introduce different conventions.

type. We rely on semantic types, which are defined using descriptions automatically identified from the syntax (lists of variations), rather than pure syntactical ones. This helps during classification of identified concepts: a syntactical concept would map "s" to "serotonin"; but additionally, we need to express that the given gene demands any arbitrary form of a reference to a substance, which is serotonin, in its name. Whether this occurs as the substance's name itself, an abbreviation, or synonym of the substance, and at which position in a text[5], then becomes less important concerning the matching strategy. Table 1 sums up some of the known types of tokens and examples we want to distinguish. Note that the proper type definition cannot automatically be assigned to a concept. Concepts can be identified as belonging to the same type only because they share certain properties (can be skipped, is a numerical entity, is a mandatory tokens that occurs at the end of a name.) In Table 1, the descriptors "antigen" and "ligand", for instance, appear to be of the same type, but analysis will reveal that while the mention of "antigen" in a name is skipped frequently, "ligand" represents a mandatory concept in many synonyms.

For the remainder of this paper, we subsequently break down a gene into the basic concepts described in one or more of its name. First, a gene is identified by a set of *names* (synonyms). Second, each name consists of multiple *parts*; proper separation and identification is a crucial step. Third, each part of a name then represents a certain *concept* that is typical for the gene. A gene is defined by all identified concepts. While a gene name part stores the information on where and if it occurs in the sequence of parts that ultimately form the (or rather a) name of the gene, concepts store information about variations. Knowledge about name parts and concepts is then transferred within each respective level only. Each such potential transfer we call a *composition rule*. An example, which we will also discuss in the next section, is the gene FASLG. Is has multiple synonyms, "FASLG" being one of those. This name can be separated into the parts "FAS" and "LG". The first part has the concept "FAS", which can appear in the variations "Fas", "fas", or "CD95", as we will see later; the second part has the concept "LG", a possible variation is "ligand":



(from top to bottom, levels depict the name, parts, concepts, and variations of each concept.)

2.3 Analysis of intra-gene variations

In this section we explain how we discover concepts and their appearances (exact tokens) within a set of synonyms under the assumption that they all belong to the same gene. Basically, this means that we can allow for more mismatches, lacking parts, or the like, as for comparing names of different genes.

Reconsider the example of the aforementioned FASLG gene (356)[6]. We show the synonyms known according to Entrez Gene in Table 2. Pairwise analysis of the synonyms provides insights as shown in Table 3.

Recombining the extracted concepts and using different variations for either, we can achieve some new potential names, for instance, FasL (capitalization) and CD95 ligand (replaced 'L' with identified

[5]Maybe within a somewhat confined neighborhood, for instance, in the current paragraph or in the abstract of the text.

[6]In the following, we will always show each gene's official symbol first and then known synonyms. Numbers in brackets refer to Entrez Gene IDs.

Apoptosis antigen ligand	APTL	apoptosis (APO-1) antigen ligand 1
Apoptosis (APO-1) ligand 1	APT1LG1	FAS antigen ligand
Apoptosis ligand	CD178	Fas ligand (TNF superfamily, member 6)
CD95L	FASL	TNFL6_HUMAN
fas ligand	FASLG	TNFSF6
FAS ligand	TNFL6	Tumor necrosis factor ligand superfamily member 6

Table 2: Synonyms of the FASLG gene that we use in our examples.

Synonyms	Composition rule learned	No.
FASL + FAS ligand	L ≡ ligand	1
FASLG + FAS ligand	LG ≡ ligand	2
FAS ligand + fas ligand	FAS ≡ fas	3
FASL + CD95L	FAS ≡ CD95	4
Tumor necrosis factor ligand superfamily member 6 + TNFSF6	T ≡ Tumor, N ≡ necrosis F ≡ factor, SF ≡ superfamily "member" before a number can be left out	5a,b 5c,d 5e
Apoptosis antigen ligand + Apoptosis ligand	"antigen" can be left out	6
FAS antigen ligand + FAS ligand	"antigen" can be left out	7
Apoptosis (APO-1) ligand 1 + Apoptosis ligand	"1" at end can be left out	8
TNFL6 + TNFL6_HUMAN	"_HUMAN" can be added to a name	9
Fas ligand (TNF superfamily, member 6) + FAS ligand	Fas ≡ FAS	10
Apoptosis ligand + APTL	Apoptosis ≡ APT	11
Apoptosis (APO-1) ligand 1 + APT1LG1	ligand 1 ≡ LG1	12

Table 3: Pairwise analysis of some synonyms for FASLG and some insights gained. Conclusions shown in the bottom part can be drawn using insights from the first part only. Rules like "X can be left out" imply that the opposite can also happen, "X can be added", and vice versa. Multiple detections of the same rule (no. 6 & 7) increase its support, so the application of rules could be weighted accordingly.

long form) for the FASLG gene. In cases where neither part of a name can be mapped onto parts of another name, then no rule should be generated: comparing "CD178 antigen" to "CD95 ligand" should not result in the variation "CD178 ligand". On the other hand, after removal of "antigen" (rules no. 6 & 7 in Table 3), "CD178" represents a variation of "CD95 ligand" (which in this case was already known from Entrez Gene.) In the following sections, we explain the detection of different kinds of variations in more detail and show examples.

Abbreviations

Detecting abbreviations is a crucial initial step in our analyses. Many variations are explained only across abbreviations and their long forms. More important, comparing abbreviations and long forms identifies the parts of either name, which can then be compared to parts of other names. Taking HIF1A (3091) as an example, we find the synonyms "HIF1 alpha", "HIF-1 alpha", "HIF-1alpha", and "Hypoxia-inducible factor 1 alpha". Schwartz and Hearst's algorithm easily reveals that "1 alpha", "1alpha", and "1A" all map to each other; "H" can be mapped to "Hypoxia", and so on. All in all, we learned that "Hypoxia-inducible factor 1A" could be a potential synonym for HIF1A.

We now look at the OR1G1 gene (8390). Consider two of its synonyms, "Olfactory receptor 1G1", and "olfactory receptor, family 1, subfamily G, member 1". Comparing the official symbol with the first synonym, it becomes clear that "OR" abbreviates "Olfactory receptor" using S&H. Comparing the synonyms, we find direct correspondences between both "1"s and "G". AS we are still within one gene, is is safe to assume that all in all, "1G1" abbreviates "family 1, subfamily G, member 1". This implies that concepts stating that we are within a gene family (subfamily, members) can be missing – whereas the respective values ("1", "G", "1") are mandatory.

Another abbreviation that commonly occurs in gene names is the (abbreviated) mention of the organism (on the species level). For example, the gene GIMAP4 (55303) has "HIMAP4", "IMAP4", "IAN1", "hIAN1", and "human immune associated nucleotide 1" as known synonyms. From synonyms 1 and 2 we can infer that an "H" can be added to a name (just like "_HUMAN", see Table 3.) The same is true for "h" (synonyms 3 and 4.) Comparing synonyms 1 or 4 to 5 leads to the conclusion that "H"

and "h" both abbreviate "human."

Lexical variations

In the set of synonyms for ADHFE1 (137872), we find "Fe-containing alcohol dehydrogenase 1" and "alcohol dehydrogenase, iron containing, 1". Splitting these synonyms into their respective parts and then comparing both sets reveals that all but one part each can be exactly mapped to a corresponding part in the other synonym. From this almost exact match, we can conclude that the parts "Fe" and "iron" are synonyms of each other, potentially representing the same concept, and easy to confirm for a human.

In the same manner, we will find that "1B" can be sometimes expressed as "2", and that "adaptor" and "Adapter" are orthographic variations of each other, by looking at some synonyms for AP1S2 (8905):

- Adapter-related protein complex 1 sigma 1B subunit
- adaptor-related protein complex 1 sigma 2 subunit
- adaptor-related protein complex 1, sigma 1B subunit

To detect these two changes, we first need to map parts to each other and then compare the names based on the sequence of the parts.

Structural variations

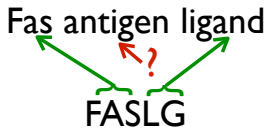
Changes in the structure of a name can be deduced when a safe mapping between most parts of a name exist. For the HMMR gene (3161), we find two evidences for such a variation, which also lead to the conclusion that "for" is an optional part. However, in our system, we would retain information concerning the positioning of "for" (at least, tendencies like "not the first" and "not the last" part.)

- Receptor for hyaluronan-mediated motility
- hyaluronan-mediated motility receptor
- Hyaluronan mediated motility receptor
- intracellular hyaluronic acid binding protein
- hyaluronan-mediated motility receptor (RHAMM)

Analysis of this example also finds that "hyaluronan" can start with an upper case letter (and that this occurs only when it is the first part of the name. "RHAMM" is the abbreviation for "Receptor for hyaluronan-mediated motility", as revealed by S&H. This leads to the next conclusion, that abbreviations can immediately follow a gene name.

Descriptive elements

Comparing the sequence of identified name parts (parts of a name where a mapping from this part to a part of the other synonym exists) yields dissimilarities that result either from a dropped/added name part, or from a lexical variation. Consider the following example:



Inexact matching immediately identifies the mapping from "Fas" to "FAS"; abbreviation detection and/or alignment yields "ligand" as a long form/variation of "LG." The sequence of name parts if the same in both synonyms, with an added "antigen" in the first synonym. An extracted composition rule could thus be that "antigen" is of additional, descriptive value only, and can be skipped. Knowing this, the first synonym should also match the strings "Fas ligand" and "FAS ligand" (in fact, both should.)

Another example is ZG24P (259291) with its synonym "uncharacterized gastric protein ZG24P". As the official symbol clearly is an abbreviation (single word, upper case letters, numbers) and matches the last part of the synonym, we can assume that the first part is either another synonym or a mere descriptive element that explains the real gene name. Indeed, patterns like "uncharacterized ... protein" or "hypothetical protein" appear frequently as first parts of gene names.

2.4 Analysis of inter-gene variations

As we have so far analyzed synonyms of one and the same gene to extract knowledge on name composition, we can now apply this knowledge to the whole set of gene names. This means, that we add knowledge gained by analyzing one gene to other genes, wherever applicable. Essentially, this comes down to finding corresponding concepts in two or more genes' names, and joining the information contained in each concept. If within one gene name it became clear that "L" and "ligand" represent the same concept, and for another gene "L" and "LG" are variations of the same concept, then a combined concept would have all three variations. The combined concept then replaces the old concepts. We apply the same idea to name parts, for which information about their ordering etc. was extracted.

Inter-gene analysis also reveals the main distinctive features of single gene names or groups of names (for instance, families.) Some names differ only in Arabic/Roman numbers or in Greek let-

ters. Potentially they belong to the same group, as different members or subtypes. Knowing how to find one family member implicitly means knowing how to find the others. Thus, it helps identify crucial parts (for the family name) and distinctive parts (for the exact member.) A matching strategy could thus try to find the family name and then look for any reference to a number. Knowledge about this kind of relationships has to be encoded in the dictionary, however. Spotting a gene family's name without any specific number could lead to the assignment of the first member to the match, see Table 3, rule no. 8 (or dismissing the name, depending on user-specific demands). Such information can also be used for disambiguating names. Analyzing the names "CD95 ligand" and "CD95 receptor" of two different genes, it can be concluded that "CD95" by itself contains not enough information to justify the identification of either gene directly. Finding other "receptor"s in the dictionary will also mark "receptor" as a concept crucial, but not sufficient, for identifying a gene's name in text. For "CD95", on the other hand, we have shown before that this token might be exchanged with others.

Knowledge about (partial) abbreviations, like in aforementioned "HIF" = "Hypoxia-inducible factor" and "OR" = "olfactory receptor", can be transferred to all synonyms from other entries in the dictionary that have the same long or short forms (but possibly do not mention the respective other in any synonym.) Similarly, presumed lexical variations ("gastric" versus "stomach") that have been found for one gene name (one concept) can be included in all corresponding concepts to spread the information that "gastric" can appear as "stomach" in text. This is necessary to detect the name "stomach alcohol dehydrogenase", where the corresponding Entrez Gene entry (ADH7, 131) does have the token "stomach" in any of its synonyms.

Also, synonyms mentioning the species (like "hIAN1" to depict human) are not contained for every entry. Learning that "h" can be added to a gene name helps recognizing such a variation in text for other names (the dictionary lacks the variation "hFasL" of FASLG, which is sometimes used.)

3 Evaluation and Conclusions

We evaluated some ideas presented in this paper on the BioCreAtIvE 2 (BC2) dataset for the gene normalization task. For the purpose of this study, we were interested in how our method would perform concerning the recall, as compared to methods based on hand-curated dictionary refinement. We conducted the following experiment: the BC2 GN gold standard consists of references to abstracts (PubMed IDs), genes identified in each abstract (Entrez Gene IDs) and text snippets that comprise each gene's name. For one abstract, there could be multiple, different snippets representing the same gene, ADH7 (131): "stomach alcohol dehydrogenase", "class IV alcohol dehydrogenase", or "sigma-ADH", all in the same abstract. For identification, it was sufficient in BC2 to report the ID, regardless of number of occurrences or name variations.

As the method presented in this paper lacks a matching strategy for spotting of names, we performed our initial evaluation on the text snippets only. Finding the right ID for *each* snippet thus ultimately yielded the recall performance. In the above example, we would try to identify ID 131 three times, counting every miss as a false negative. The methods presented above were able to yield a recall of 73.1%. With the original BC2 evaluation scheme, we achieve a recall of 84.2%. Compared to the highest result for our system with a manually refined dictionary, this figure is more than 8% lower. This shows that still, many name variations are not recognized. Some errors could be accounted to ranges or enumerations of gene names ("SMADs 1, 5 and 8"), others to not far enough reaching analyses: for detecting "SMAD8", we only had the synonyms "SMAD8A", "SMAD8B", and "SMAD9" for the correct gene in the dictionary (all are synonyms for the same gene, according to Entrez Gene). It should thus have been learned that the letter "A" can be left out (similar to "1", see rule no. 8 in Table 3.) Another undetected example is "G(olf) alpha" (GNAL, 2774). Rules to restrict either of the synonyms

- Guanine nucleotide-binding protein G(olf), alpha subunit
- guanine nucleotide binding protein (G protein), alpha stimulating activity polypeptide, olfactory type
- Adenylate cyclase-stimulating G alpha protein, olfactory type
- Guanine nucleotide-binding protein, alpha-subunit, olfactory type

to this mentioning in text could have been deduced as follows:

(1) Learn in another gene: description before "protein" can be left out ⇒ "G(olf), alpha subunit" could be a name of its own.

(2) Learn in this or another gene: "alpha subunit" can be expressed as "alpha" (or "subunit" skipped) ⇒ "G(olf) alpha" could be a name.

We see that most orthographical and morphological variations (Greek symbols/English words, singular/plural forms, capitalization) can be integrated quite easily in matching techniques. The general knowledge about such variations is far-reaching and can be applied to most domains. In contrast, structural and lexical variations are much harder to pinpoint and express in general ways; mostly, such possible variations are specific to a sub-domain and thus present the main challenge for our method.

The ideas discussed in this paper originated from work on the aforementioned BioCreAtIvE 2 task. In that work, we used manually designed rules to generate variations of gene names. Hanisch et al. (Hanisch et al., 2005) and other groups propose similar methods all based on human observation and experience leading to refined dictionaries. As many causes for name variations are easy to spot and express, we concluded it was entirely possible to gain such insights in an automated manner. Left undetermined is the potential impact of composition rules on machine-learning techniques that use dictionaries as input for features.

However, the methodology should work for other task using the same or similar initial observations (This remains to be proven.) We are currently applying the method to the analysis of Gene Ontology terms (Ashburner et al., 2000). There, many terms are mere descriptions of concepts than precise labels, and there are less additional synonyms (with structural and lexical variations.) A good starting point for assessing possible patterns in name composition could also be the MeSH controlled vocabulary. Entries in MeSH typically contain many structural and lexical variations, a deeper understanding of which bears more insights than of orthographical or morphological variations.

Readers of this manuscript should either gain more insights into name compositions of gene names –in order to help refining dictionaries based on manual rule sets–, or be convinced that the idea of learning composition rules can be tackled in automated ways, promising examples of and basic techniques for which we discussed herein.

Supplementary information

The extracted set of rules for name variations and an extended dictionary for human genes, originating from Entrez Gene, are available at http://www.informatik.hu-berlin.de/~hakenber/publ/suppl/ . The dictionary can directly be used for matching entries against text and covers 32,980 genes. The main Java classes are available on request from the authors.

References

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, et al. 2000. Gene Ontology: Tool for the Unification of Biology. *Nature Genetics*, 25:25–29.

Aaron M. Cohen and William R. Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71.

Jeremiah Crim, Ryan McDonald, and Fernando Pereira. Automatically annotating documents with normalized gene lists. 2005. *BMC Bioinformatics*, 6(Suppl 1):S13.

Jörg Hakenberg, Steffen Bickel, Conrad Plake, Ulf Brefeld, Hagen Zahn, Lukas Faulstich, Ulf Leser, and Tobias Scheffer. 2005. Systematic feature evaluation for gene name recognition. *BMC Bioinformatics*, 6(Suppl 1):S9.

Jörg Hakenberg, Loic Royer, Conrad Plake, Hendrik Strobelt. 2007. Me and my friends: gene mention normalization with background knowledge. *Proc 2nd BioCreative Challenge Evaluation Workshop*, April 23-25 2007, Madrid, Spain.

Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, Juliane Fluck ProMiner: rule-based protein and gene entity recognition. 2005. *BMC Bioinformatics*, 6(Suppl 1):S14.

Ulf Leser and Jörg Hakenberg. 2005. What Makes a Gene Name? Named Entity Recognition in the Biomedical Literature. *Briefings in Bioinformatics*, 6(4):357–369.

Donna Maglott, Jim Ostell, Kim D. Pruitt, and Tatiana Tatusova. 2005. Entrez Gene: gene–centered information at NCBI. *Nucleic Acids Research*, 33(Database Issue):D54–D58.

Alexander Morgan and Lynette Hirschman. 2007. Overview of BioCreative II Gene Normalization. In: *Proc 2nd BioCreative Challenge Evaluation Workshop*, April 23-25 2007, Madrid, Spain.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–53.

Ariel S. Schwartz and Marti A. Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. *Proc Pac Sym Bio*, 451–462.

Hua Xu, Jung-Wei Fan, George Hripcsak, Eneida A. Mendonça, Marianthi Markatou, and Carol Friedman. 2007. Gene symbol disambiguation using knowledge-based profiles. *Bioinformatics*, 23(8):1015–1022.

Exploring the Use of NLP in the Disclosure of Electronic Patient Records

David Hardcastle
Faculty of Mathematics and Computing
The Open University
d.w.hardcastle@open.ac.uk

Catalina Hallett
Faculty of Mathematics and Computing
The Open University
c.hallett@open.ac.uk

Abstract

This paper describes a preliminary analysis of issues involved in the production of reports aimed at patients from Electronic Patient Records. We present a system prototype and discuss the problems encountered.

1 Introduction

Allowing patient access to Electronic Patient Records (EPR) in a comprehensive format is a legal requirement in most European countries. Apart from this legal aspect, research shows that the provision of clear information to patients is instrumental in improving the quality of care (Detmer and Singleton, 2004). Current work on generating explanations of EPRs to patients suffer from two major drawbacks. Firstly, existing report generation systems have taken an intuitive approach to the generation of explanation: there is no principled way of selecting the information that requires further explanation. Secondly, most work on medical report generation systems has concentrated on explaining the structured part of an EPR; there has been very little work on providing automatic explanations of the narratives (such as letters between health practitioners) which represent a considerable part of an EPR. Attempting to rewrite narratives in a patient-friendly way is in many ways more difficult than providing suggestions for natural language generation systems that take as input data records. In narratives, ambiguity can arise from a combination of aspects over which NLG systems have full control, such as syntax, discourse structure, sentence length, formatting and readability.

This paper introduces a pilot project that attempts to address this gap by addressing the following research questions:

1. Given the text-based part of a patient record, which segments require explanation before being released to patients?
2. Which types of explanation are appropriate for various types of segment?
3. Which subparts of a segment require explanation?

The prototype system correctly selects the segments that require explanation, but we have yet to solve the problem of accurately identifiying the features that contribute to the "expertness" of a document. We discuss the underlying issues in more detail in section 3 below.

2 Feature identification method

To identify a set of features that differentiate medical expert and lay language, we compared a corpus of expert text with a corpus of lay texts. We then used the selected features on a corpus of narratives extracted from a repository of Electronic Patient Records to attempt to answer the three questions posed above. First, paragraphs that contain features characteristic to expert documents are highlighted using a corpus of patient information leaflets as a background reference. Second, we prioritise the explanations required by decomposing the classification data. Finally, we identify within those sections the features that contribute to the classification of the section as belonging to the expert register, and provide suggestions for text simplification.

2.1 Features

The feature identification was performed on two corpora of about 200000 words each: (a) an expert corpus, containing clinical case studies and medical manuals produced for doctors and (b) a lay corpus, containing patient testimonials and informational materials for patients. Both corpora were

BioNLP 2007: Biological, translational, and clinical language processing, pages 161–162,
Prague, June 2007.

sourced from a variety of online sources. In comparing the corpora we considered a variety of features in the following categories: medical content, syntactic structure, discourse structure, readability and layout. The features that proved to be best discriminators were the frequency of medical terms, readability indices, average NP length and the relative frequency of loan words against English equivalents[1]. The medical content analysis is based on the MeSH terminology (Canese, 2003) and consists of assessing: (a) the frequency of MeSH primary concepts and alternative descriptions, (b) the frequency of medical terms types and occurences and (c) the frequency of MeSH terms in various top-level categories. The readability features consist of two standard readability indices (FOG and Flesch-Kincaid). Although some discourse and layout features also proved to have a high discriminatory power, they are strongly dependent on the distribution medium of the analysed materials, hence not suitable for our analysis of EPR narratives.

2.2 Analysing EPR narratives

We performed our analysis on a corpus of 11000 narratives extracted from a large repository of Electronic Patient Records, totalling almost 2 million words. Each segment of each narrative was then assessed on the basis of the features described above, such as Fog, sentence length, MeSH primary concepts etc. We then smoothed all of the scores for all segments for each feature forcing the minimum to 0.0, the maximum to 1.0 and the reference corpus score for that feature to 0.5. This made it possible to compare scores with different gradients and scales against a common baseline in a consistent way.

3 Evaluation and discussion

We evaluated our segment identification method on a set of 10 narratives containing 27 paragraphs, extracted from the same repository of EPRs . The segment identification method proved succesful, with 26/27 (96.3%) segments marked correctly are requiring/not requiring explanation. However, this only addresses the first of the three questions set out above, leaving the following research questions open to further analysis.

Quantitative vs qualitative analysis
Many of the measures that discriminate expert from lay texts are based on indicative features; for example complex words are indicative of text that is difficult to read. However, there is no guarantee that individual words or phrases that are indicative are also representative - in other words a given complex word or long sentence will contribute to the readability score of the segment, but may not itself be problematic. Similarly, frequency based measures, such as a count of medical terminology, discriminate at a segment level but do not entail that each occurrence requires attention.

Terminology
We used the MeSH terminology to analyse medical terms in patient records, however (as with practically all medical terminologies) it contains many non-expert medical terms. We are currently investigating the possibility of mining a list of expert terms from MeSH or of making use of medical-lay aligned ontologies.

Classification
Narratives in the EPR are written in a completely different style from both our training expert corpus and the reference patient information leaflets corpus. It is therefore very difficult to use the reference corpus as a threshold for feature values which can produce good results on the corpus of narratives, suggesting that a statistical thresholding technique might be more effective.

Feature dependencies
Most document features are not independent. Therefore, the rewriting suggestions the system provides may themselves have an unwanted impact on the rewritten text, leading to a circular process for the end-user.

[1] An in-depth analysis of unfamiliar terms in medical documents can be found in (Elhadad, 2006)

References

Kathi Canese. 2003. New Entrez Database: MeSH. *NLM Technical Bulletin*, March-April.

D. Detmer and P. Singleton. 2004. The informed patient. Technical Report TIP-2, Judge Institute of Management, University of Cambridge, Cambridge.

Noemi Elhadad. 2006. Comprehending technical texts: Predicting and defining unfamiliar terms. In *Proceeding of AMIA'06*, pages 239–243.

BaseNPs that contain gene names: domain specificity and genericity

Ian Lewin
Computer Laboratory
University of Cambridge
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
ian.lewin@cl.cam.ac.uk

Abstract

The names of *named entities* very often occur as constituents of larger noun phrases which denote different types of entity. Understanding the structure of the embedding phrase can be an enormously beneficial first step to enhancing whatever processing is intended to follow the named entity recognition in the first place. In this paper, we examine the integration of general purpose linguistic processors together with domain specific named entity recognition in order to carry out the task of baseNP detection. We report a best F-score of 87.17% on this task. We also report an inter-annotator agreement score of 98.8 Kappa on the task of baseNP annotation of a new data set.

1 Introduction

Base noun phrases (baseNPs), broadly "the initial portions of non-recursive noun phrases up to the head" (Ramshaw and Marcus, 1995), are valuable pieces of linguistic structure which minimally extend beyond the scope of named entities. In this paper, we explore the integration of different techniques for detecting baseNPs that contain a named entity, using a domain-trained named entity recognition (NER) system but in combination with other linguistic components that are "general purpose". The rationale is simply that domain-trained NER is clearly a necessity for the task; but one might expect to be able to secure good coverage at the higher syntactic level by intelligent integration of general purpose syntactic processing without having to undergo a further round of domain specific annotation and training. We present a number of experiments exploring different ways of integrating NER into general purpose linguistic processing. Of course, good results can also be used subsequently to help reduce the effort required in data annotation for use in dedicated domain-specific machine learning systems for baseNP detection.

First, however, we motivate the task itself. Enormous effort has been directed in recent years to the automatic tagging of named entities in bio-medical texts and with considerable success. For example, iHOP reports gene name precision as being between 87% and 99% (depending on the organism) (Hoffman and Valencia, 2004). Named entities are of course only sometimes identical in scope with noun phrases. Often they are embedded within highly complex noun phrases. Nevertheless, the simple detection of a name by itself can be valuable. This depends in part on the intended application. Thus, iHOP uses gene and protein names to hyperlink sentences from Medline and this then supports a browser over those sentences with additional navigation facilities. Clicking on *Dpp* whilst viewing a page of information about *hedgehog* leads to a page of information about *Dpp* in which sentences that relate both *Dpp* and *hedgehog* are prioritized.

One of the application advantages of iHOP is that the discovered gene names are presented to the user in their original context and this enables users to compensate for problems in reliability and/or contextual relevance. In many Information Extraction (IE) systems, relations between entities are detected and extracted into a table. In this case, since the im-

BioNLP 2007: Biological, translational, and clinical language processing, pages 163–170,
Prague, June 2007.

mediate surrounding context of the gene name may be simply lost, the reliability of the original identification becomes much more important. In section 2 below, we explain our own application background in which our objective is to increase the productivity of human curators whose task is to read particular scientific papers and fill in fields of a database of information about genes. Directing curators' attention to sentences which contain gene names is clearly one step. Curators additionally report that an index into the paper that uses the gene name and its embedding baseNP is even more valuable (reference omitted for anonymity). This often enables them to predict the possible relevance of the name occurrence to the curation task and thus begin ordering their exploration of the paper. Consequently, our technical goal of baseNP detection is linked directly to a valuable application task. We also use the baseNP identification in order to type the occurrence semantically and use this information in an anaphora resolution process (Gasperin, 2006).

The detection of baseNPs that contain a named entity is a super-task of NER, as well as a sub-task of NP-chunking. Given that NER is clearly a domain specific task, it is an interesting question what performance levels are achievable using domain trained NER in combination with general purpose linguistic processing modules.

There is a further motivation for the task. The distinction between a named entity and an embedding noun phrase is one with critical importance even for the sub-task of NER. Dingare et al (2005) conclude, from their analysis of a multi-feature maximum entropy NER module, that increases in performance of biomedical NER systems will depend as much upon qualitative improvements in annotated data as in the technology underlying the systems. The claim is that quality problems are partly due to confusion over what lies in the scope of a named entity and what lies at higher syntactic levels. Current biomedical annotations are often inconsistent partly because annotators are left with little guidance on how to handle complexities in noun phrases, especially with respect to premodifiers and conjunctions. For example, which premodifiers are part of the *named entity* and which are "merely" part of the embedding noun phrase? Is *human* part of the named entity in *the regulation of human interleukin-2 gene expression*, or not?

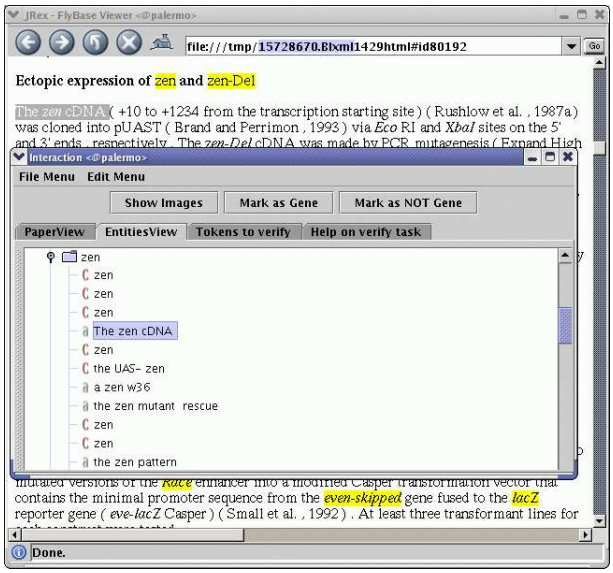


Figure 1: Paper Browser showing baseNP index

By focussing attention instead on the baseNPs that contain a named entity, one can clearly sidestep this issue to some extent. After all, increasing the accuracy of an NER module with respect to premodifier inclusion is unlikely to affect the overall accuracy of detection of the embedding noun phrases.

2 FlyBase curation

The intended application for our work is a software environment for FlyBase curators that includes an NLP-enhanced Browser for Scientific Papers. FlyBase is the world's leading genomics database for the fruitfly Drosophila (melanogaster and other species) (Crosby et al., 2007). FlyBase is largely updated through a paper-by-paper methodology in which research articles likely to contain information relevant for the FlyBase database are first put in a priority list. Subsequently, these are read by skilled geneticists (at post-doctoral level) who distil gene related information into the database itself. Although this is a paradigm example of IE, our objective is not to fully automate this task itself, simply because the expected accuracy rates are unlikely to be high enough to provide a genuinely useful tool. Rather, our task is to enable curators to explore the gene related sections of papers more efficiently. The Browser currently highlights potential

items of interest for curators and provides novel indexing and navigation possibilities. It is in this context that the identification of baseNPs that contain gene names is carried out. An individual sentence that contains a gene name is very often not enough, considered in isolation, for curators to fill in a required database field. Information often needs to be gathered from across a paragraph and even the whole paper. So extraction of sentences is not an attractive option. Equally, a whole sentence is unfeasibly large to serve simply as an indexing term into the paper. Noun phrases provide more information than simply gene names, but post-modification can also lead to extremely long terms. BaseNPs are therefore a useful compromise, these being short enough to display whole in a window (i.e. no scrolling is required) and often bearing enough information for the user to understand much more of the context in which the gene name itself appears. Furthermore, the baseNP is both a natural "unit" of information (whereas a window of n tokens around a gene name is not) and it supports further processing. BaseNPs are typed according to whether they denote genes or various gene products and linked together in anaphoric chains.

In our navigation panel for the Browser, the baseNPs are sorted according to the gene name that they contain (and then by order in which they appear within the paper), and hyperlinked to their occurrence in the paper. This enables users to explore papers gene-by-gene but also, when considering a particular gene, to understand more about the reference to the gene - for example whether gene products or promoters are being referenced. Figure 1 contains an example screenshot.

3 Scope of the Data

Complex nominals have long been held to be a common feature in scientific text. The corpus of Vlachos and Gasperin (2006) contains 80 abstracts (600 sentences) annotated with gene names. In this data-set, noun phrases that contain gene names (excluding post-modifiers) of 3 words or more comprise more than 40% of the data and exhibit primarily: strings of premodifiers *tudor mutant females*, *zygotic Dnop5 expression*; genitives: *Robo 's cytoplasmic domain*, *the rdgB protein 's amino terminal 281 residues*; coordination *the copia and mdg-1 elements* and parenthetical apposition *the female-specific gene Sex lethal (Sxl)*, and *the SuUR (suppressor of underreplication) gene*. Only 41% of the baseNPs containing a gene name consist of one token only. 16% have two tokens. The two token baseNPs include large numbers of combinations of gene names with more general words such as *Ras activity*, *vnd mutants*, *Xiro expression*, *IAP localization* and *vasa protein*. In general, the gene name appears in modifier position although species modifiers are common, such as *Drosophila Tsg*, and there are other possibilities: *truncated p85*.

Our intention is to categorize this data using the concept of "baseNP" and build effective computational models for recognizing instances. Although baseNP is a reasonably stable linguistic concept, its application to a new data-set is not completely straightforward. Ramshaw and Marcus (1995) state that a baseNP aims "to identify essentially the initial portions of nonrecursive noun phrases up to the head, including determiners but not including postmodifying prepositional phrases or clauses". However, work on baseNPs has essentially always proceeded via algorithmic extraction from fully parsed corpora such as the Penn Treebank. BaseNPs have therefore depended on particular properties of the annotation framework and this leads to certain aspects of the class appearing unnatural.

The clearest case is single element conjunction, which Penn Treebank policy dictates is annotated at word-level with a flat structure like this *[lpl and xsl]* (brackets indicate baseNP boundaries). As soon as one of the elements is multi-word however, then separate structures are to be identified *[lpl] and [the sxl gene]*. The dependency on numbers of tokens becomes clearly problematic in the bio-medical domain. Quite different structures will be identified for *lpl and fasciclin*, *lpl and fasciclin 1* and possibly *lpl and fasciclin-1*, depending on how tokenization treats hyphens. Furthermore, nothing here depends on the motivating idea of "initial segments up to the head". In order to provide a more natural class, our guidelines are that unless there is a shared modifier to account for (as in *[embryonic lgl and sxg]*), all coordinations are split into separate baseNPs. All other cases of coordination follow the standard guidelines of the Penn Treebank.

A second difficult case is possessives. BaseNP extraction algorithms generally split possessives like this: [fra] ['s ectodomain], corresponding (somewhat) to an intuition that there are two NPs whilst assigning each word to some baseNP chunk and not introducing recursiveness. This policy however causes a sharp division between this case and *the fra ectodomain* following the Penn Treebank bracketing guideline that nominal modifiers are never labelled. Since our interest is "the smallest larger NP containing a gene name", we find it much more natural to treat *fra's* as just another modifier of the head *ectodomain*. Whether it recursively contains a single word NP *fra* (or just a single word NNP) is again not something that is motivated by the idea of "initial segments up to the head". Similarly, we mark one baseNP in *the rdgB protein's amino terminal 281 residues*, viz. *the rdgB protein*.

Apposition, as in *Sex lethal (Sxl)* and *the gene sex lethal* , is a further interesting case. In the first case, "Sex lethal" and "Sxl" stand in apposition. Both are gene names. The former is the head. In the second, "gene" is the head and "sex lethal" is a name that stands in apposition. In each case, we have a head and post-modifiers which are neither clausal nor prepositional. It is unclear whether the rubric "clausal or prepositional" in Ramshaw and Marcus' statement of intent is merely illustrative or definitive. On the grounds that a sharp division between the non-parenthetical case *the gene sex lethal* and the pre-modifier *the sex lethal gene* is unnatural, our intuition is that the baseNP does cover all 4 tokens in this case. All (post-head) parentheticals are however to be treated more like optional adjuncts and therefore not included with the head to which they attach.

In order to verify the reliability of baseNP annotation, two computational linguists (re)annotated the 600 sentences (6300 tokens) of Vlachos and Gasperin (2006) with baseNPs and heads using the published guidelines. We added material concerning head annotation. Vlachos and Gasperin did not quote agreement scores for baseNP annotation. Their interest was directed at gene name agreement between a linguist and a biologist. Our 2-person inter-annotator Kappa scores were 0.953 and 0.988 on head and baseNP annotation respectively representing substantial agreement.[1] .

4 Methodology

A reasonable and simple baseline system for extracting baseNPs that contain a gene name is to use an off-the-shelf baseNP extractor and simply filter the results for those that contain a gene name. To simplify analysis of results, except where otherwise noted this filter and subsequent uses of NER are based on a gold standard gene name annotation. In this way, the contributions of different components can be compared without factoring in relative errors of NER. Naturally, in the live system, an automated NER process is used (Vlachos and Gasperin, 2006). For the baseline we chose an implementation of the Ramshaw and Marcus baseNP detector distributed with GATE[2] pipelined with the Stanford maximum entropy part of speech tagger [3]. The Stanford tagger is a state of the art tagger incorporating a number of features including use of tag contexts, lexical features, a sophisticated smoothing technique, and features for unknown words (including 4-gram prefixes and suffixes). Both components of the baseline systems utilize the 48 tag Penn Treebank tagset. Results however showed that poor performance of the part of speech tagger could have a disastrous effect on baseNP detection. A simple extension of the baseline is to insert a module in between POS tagging and NP detection. This module revises the POS tags from the tagger in the light of NER results, essentially updating the tags of tokens that are part of named entities. This is essentially a simple version of the strategy mooted by Toutanova at el (2003) that the traditional order of NER and tagging be reversed. It is simpler because, in a maximum entropy framework, NER results can function as one extra feature amongst many in POS detection; whereas here it functions merely as an override. Retraining the tagger did not form part of our current exploration.

[1] In fact, although the experiment can be considered a classification of 6300 tokens in IOB format, the counting of classifications is not completely straightforward. The task was "annotate the baseNP surrounding each gene name" rather than "annotate each token". In principle, each token is examined; in practice a variable number is examined. If we count all tokens classified into NPs plus one token of context either side, then both annotators annotated over 930 tokens.

[2] http://www.gate.ac.uk

[3] http://nlp.stanford.edu/software/tagger.shtml

We adopted a similar strategy with the domain independent full parsing system RASP (Briscoe et al., 2006). RASP includes a simple 1st order HMM POS tagger using 149 of the CLAWS-2 tagset. The tagger is trained on the manually corrected subsets of the (general English) Susanne, LOB and BNC corpora. The output of the tagger is a distribution of possible tags per token (all tags that are at least 1/50 as probable as the top tag; but only the top tag if more than 90% probable). The tagger also includes an unknown word handling module for guessing the possible tags of unknown words. The RASP parser is a probabilistic LALR(1) parser over the CLAWS-2 tags, or, more precisely, a unification grammar formalism whose lexical categories are feature based descriptions of those tags. The parser has no access to lexical information other than that made available by the part of speech tags. Although the output of RASP is a full parse (or a sequence of fragments, if no connected parse can be found) and baseNPs may not be constituents of NPs, baseNPs can be extracted algorithmically from the full parse.

Some more interesting pre-parsing integration strategies are available with RASP because it does not demand a deterministic choice of tag for each word. We experimented with both a deterministic re-write strategy (as for the baseline system) and with various degrees of interpolation; for example, adjusting the probability distribution over tags so that proper noun tags receive 50% of the probability mass if the token is recognized by NER, and the other tags receive the remaining 50% in direct proportion to the amount they would receive from the POS tagger alone. In this set-up, the NER results need not function simply as an override, but equally they do not function simply as a feature for use in part of speech tagging. Rather, the parser may be able to select a best parse which makes use of a sequence of tags which is not itself favoured by the tagger alone. This allows some influence to the grammatical context surrounding the gene name and may also permit tags within phrasal names such as *transforming growth factor* to propagate.

RASP is also a non-deterministic parser and consequently a further possible integration strategy is to examine the output n-best list of parses to find baseNPs, rather than relying on simply the 1-best output. The n-best parses are already scored according to a probabilistic model trained on general text. Our strategy is to re-score them using the additional knowledge source of domain specific NER. We explored a number of re-scoring hypotheses. First, a cut-off of 20 on n-best lists was found to be optimal. That is, correct analyses tended to either be in the top 20 or else not in the top 100 or even 1000. Secondly, differences in score between the incorrect 1-best and the correct nth hypothesis were not a very reliable indicator of "almost right". This is not surprising as the scores are probabilities calculated over the complete analysis, whereas our focus is one small part of it. Consequently, the re-scoring system uses the probabilistic model just to generate the top 20 analyses; and those analyses are then re-scored using 3 features. Analyses that concur with NER in having a named entity within an NP receive a reward of +1. Secondly, NP analyses that contain N+1 genes (as in a co-ordination) receive a score of +N, so long as the NP is single headed. For example, "gurken or torpedo females" will receive a preferred analysis in which "gurken" and "torpedo" are both modifiers of "females". The "single headedness" constraint rules out very unlikely NP analyses that the parser can return as legal possibilities. Finally, analyses receive a score of -1 if the NP contains a determiner but the head of the NP is a gene name. The top 20 parses may include analyses in which, for example, "the hypothesis that phenylalanine hydroxylase" contains "that phenylalanine hydroxylase" as an NP constituent.

Finally, we also experimented with using both the full parsing and shallow baseNP spotter together; here, the idea is simply that when two analyses overlap, then the analysis from full parsing should be preferred on the grounds that it has more information available to it. However, if the shallow spotter detects an analysis when full parsing detects none then this is most likely because full parsing has been led astray rather than it has discovered a more likely analysis not involving any baseNP.

5 Experimental Results

Table 1 gives the precision, recall and (harmonic) F-score measures for the baseline NP system with and without the extra pre-parsing retagging module; and table 2 gives similar figures for the generic full pars-

ing system. Scores for the left boundary only, right boundary only and full extent ('correct') are shown. The extra retagging module (i.e. override tagger results, given NER results) improves results in both systems and by similar amounts. This is nearly always on account of gene names being mis-tagged as verbal which leads to their exclusion from the set of baseNP chunks. The override mechanism is of course a blunt instrument and only affects the tags of tokens within gene names and not those in its surrounding context.

Table 3 shows the results from interpolating the POS tag distribution P with the NER distribution N linearly using different levels of λ. For example, $\lambda = 1.00$ is the simple retagging approach in which all the probability is assigned to the NER suggested tag; whereas $\lambda = 0.25$ means that only 25% is allocated by NER. The figures shown are for one variant of the full parsing system which included n-best selection but other variants showed similar behaviour (data not shown). The results from interpolation show that the extra information available in the parse does not prove valuable overall. Decreasing values of λ lead to decreases in performance. These results can be interpreted as similar in kind to Charniak et al (1996) who found that a parser using multiple POS tag inputs could not improve on the *tag accuracy* of a tagger outputting single POS tags. Our results differ in that the extra tag possibilities are derived from an alternative knowledge source and our measurement is baseNP detection. Nevertheless the conclusion may be that the best way forward here is a much tighter integration between NER and POS tagging itself.

POS tagging errors naturally affect the performance of both shallow and full parsing systems, though not necessarily equally. For example, the tagger in the shallow system tags *ectopic* as a verb in *vnd-expression leads to ectopic Nk6 expression* and this is not corrected by the retagging module because *ectopic* is not part of the gene name. Consequently the baseNP spotter is led into a left boundary error. Nevertheless, the distribution of baseNPs from the two systems do appear to be complementary in a rather deeper fashion. Analysis of the results indicates that parentheticals in pre-modifier positions appears to throw the shallow parser severely off course. For example, it generates the analysis

	R	P	F
retag+shallow			
(correct)	80.21	75.92	78.01
(left b)	92.40	87.46	89.86
(right b)	90.81	85.95	88.32
shallow only			
(correct)	74.03	76.32	75.16
(left b)	84.28	86.89	85.56
(right b)	82.69	85.25	83.95

Table 1: Generic shallow parsing

	R	P	F
retag+full			
(correct)	80.92	84.81	82.82
(left b)	85.69	89.81	87.70
(right b)	88.69	92.96	90.78
full only			
(correct)	75.44	85.23	80.04
(left b)	80.21	90.62	85.10
(right b)	82.51	93.21	87.54

Table 2: Generic full parsing

[the transforming growth factor-beta] ([TGF-beta]) superfamily. Also, appositions such as *the human auto antigen La* and *the homeotic genes abdominal A and abdominal B* cause problems. In these kinds of case, the full parser detects the correct analysis. On the other hand, the extraction of baseNPs from grammatical relations relies in part on the parser identifying a head correctly (for example, via a non-clausal subject relation). The shallow parser does not however rely on this depth of analysis and may succeed in such cases. There are also cases where the full parser fails to detect any analysis at all.

System	(correct)	(left b)	(right b)
λ=0.25	83.97	88.34	90.71
λ=0.50	84.16	88.69	91.22
λ=0.80	85.18	89.67	91.28
λ=1.00	85.38	89.87	91.66

Table 3: F-scores for baseNP detection for various λ

Table 4 indicates the advantages to be gained in n-best selection. The entries for *full* and *retag+full* are repeated from table 2 for convenience. The entries

System	R	P	F
retag+full	80.92	84.81	82.82
retag+full+sel	83.22	87.22	85.17
retag+full+oracle	85.87	90.17	87.96
full	75.44	85.23	80.04
full+sel	78.80	86.60	82.52
full+oracle	81.63	89.88	85.56

Table 4: Effects of n-best selection

for *full+sel* and *retag+full+sel* show the effect of adding n-best selection. The entries for *full+oracle* and *retag+full+oracle* show the maximum achievable performance by replacing the actual selection policy with an oracle that always chooses the correct hypothesis, if it is available. The results are that, regardless of whether a retagging policy is adopted, an oracle which selects the best analysis can achieve an error reduction of well over 25%. Furthermore, the simple selection policy outlined before succeeds in achieving almost half the possible error reduction available. This result is particularly interesting because it demonstrates that the extra knowledge source available in this baseNP detection task (namely NER) can profitably be brought to bear at more than one stage in the overall processing pipeline. Even when NER has been used to improve the sequence of POS tags given to the parser, it can profitably be exploited again when selecting between parses.

The complementary nature of the two systems is revealed in Table 5 which shows the effects of integrating the two parsers. baseNPs from the shallow parser are accepted whenever it hypothesizes one and there is no competing overlapping baseNP from the full parser. Note that this is rather different from the standard method of simply selecting between an analysis from the one parser and one from another. The success of this policy reflects the fact that there remain several cases where the full parser fails to deliver "apparently" simple baseNPs either because the tagger has failed to generate a suitable hypothesis, or because parsing itself fails to find a good enough analysis in the time available to it.

Overall, the best results (87.17% F-score) are obtained by applying NER results both before parsing through the update of POS tags and after it in selection from n-best lists; and by combining the results of both full parsing in order to improve analysis of more complex structures and shallow parsing as a back-off strategy. The same strategy applied using our automated gene name recognizer results in a F-score of 73.6% F-score, which is considerably less of course, although the gene name recognizer itself operates at 82.5% F-Score, with similar precision and recall figures. This naturally limits the possible performance of our baseNP recognition task. Encouragingly, the "lost" performance (just under 11%) is actually less in this scenario than when gene name recognition is perfect.

System	R	P	F
1-best	85.69	84.35	85.01
n-best	87.63	86.71	87.17
oracle	90.28	89.49	89.89

Table 5: Combining shallow and full parsing

6 Previous Work

The lack of clarity between noun phrase extents and named entity extents and its impact on evaluation and training data for NER has been noted previously, e.g. for proteins (Mani et al., 2005). Vlachos and Gasperin (2006) claim that their name versus mention distinction was helpful in understanding disagreements over gene name extents and this led, through greater clarity of *intended* coverage, to improved NER. BaseNP detectors have also been used more directly in building NER systems. Yamamoto et al (2003) describe an SVM approach to protein name recognition, one of whose features is the output of a baseNP recognizer. BaseNP recognition supplies a top-down constraint for the search for protein names within a baseNP. A similar approach albeit in a CRF framework is described in Song et al. (2005).

The concept of baseNP has undergone a number of revisions (Ramshaw and Marcus, 1995; Tjong Kim Sang and Buchholz, 2000) but has previously always been tied to extraction from a more completely annotated treebank, whose annotations are subject to other pressures than just "initial material up to the head". To our knowledge, our figures for inter-annotator agreement on the baseNP task itself

(i.e. not derived from a larger annotation task) are the first to be reported. Quality measures can be indirectly inferred from a treebank complete annotation, but baseNP identification is probably a simpler task. Doddington et al (2004) report an "overall value score of 86" for inter-annotator agreement in ACE; but this is a multi-component evaluation using a complete noun phrase, but much else besides.

Improving results through the combination of different systems has also been a topic of previous work in baseNP detection. For example, Sang et al (2000) applied majority voting to the top five machine learning algorithms from a sample of seven and achieved a baseNP recognition rate that exceeded the recognition rates of any of the individual methods.

7 Conclusion

We have motivated the task of detecting baseNPs that contain a given named entity as a task both of interest from the standpoint of use within a particular application and on more general grounds, as an intermediate point between the task of general NP chunking and domain specific NER.

We have explored a variety of methods for undertaking baseNP detection using only domain specific NER in addition to otherwise general purpose linguistic processors. In particular, we have explored both shallow and full parsing general purpose systems and demonstrated that the domain specific results of NER can be applied profitably not only at different stages in the language processing pipeline but also more than once. The best overall recognition rates were obtained by a combination of both shallow and full parsing systems with knowledge from NER being applied both before parsing, at the stage of part of speech detection and after parsing, during parse selection.

References

E.J. Briscoe, J. Carroll, and R. Watson. 2006. The second release of the rasp system. *Proc. Coling/ACL 2006 Interactive Sessions*.

E. Charniak, G. Carroll, J. Adcock, A.R. Cassandra, Y. Gotoh, J. Katz, M.L. Littman, and J. McCann. 1996. Taggers for parsers. *Artificial Intelligence*, 85(1-2):45–57.

M.A. Crosby, Goodman J.L., Strelets V.B., P. Zhang, W.M. Gelbart, and the FlyBase Consortium. 2007. Flybase: genomes by the dozen. *Nucleic Acids Research*, 35:486–491.

Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, and Claire Grover. 2005. A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations:. *Comp. Funct. Genomics*, 6(1-2):77–85.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. Automatic content extraction (ace) program - task definitions and performance measures. In *Proceedings of LREC 2004*.

C. Gasperin. 2006. Semi-supervised anaphora resolution in biomedical texts. In *Proceedings of BIONLP in HLT-NAACL06, New York*, pages 96–103.

R. Hoffman and A. Valencia. 2004. A gene network for navigating the literature. *Nature Genetics*, 36:664.

I. Mani, Z. Hu, S.B. Jang, K. Samuel, M. Krause, J. Phillips, and C.H. Wu. 2005. Protein name tagging guidelines: lessons learned. *Comparative and Functional Genomics*, 6(1-2):72–76.

L.A. Ramshaw and M.P Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 82–94. ACL.

Y. Song, G. Kim, E. ad Lee, and B. Yi. 2005. Posbiotmner: a trainable biomedical named-entity recognition system. *Bioinformatics*, 21(11):2794–2796.

E.F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*.

Erik F. Tjong Kim Sang, Walter Daelemans, Hervé Déjean, Rob Koeling, Yuval Krymolowski, Vasin Punyakanok, and Dan Roth. 2000. Applying system combination to base noun phrase identification. In *COLING 2000*, pages 857–863. Saarbruecken, Germany.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part of speech tagging with a cyclic dependency network. In *HLT-NAACL*, pages 252–259.

A. Vlachos and C. Gasperin. 2006. Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of BIONLP in HLT-NAACL06, New York*.

K. Yamamoto, T. Kudo, T. Konagaya, and Y. Matsumoto. 2003. Protein name tagging for biomedical annotation in text. In *ACL 2003 Workshop on NLP in Biomedicine*.

Challenges for extracting biomedical knowledge from full text

Tara McIntosh
School of IT
University of Sydney
NSW 2006, Australia
tara@it.usyd.edu.au

James R. Curran
School of IT
University of Sydney
NSW 2006, Australia
james@it.usyd.edu.au

Abstract

At present, most biomedical Information Retrieval and Extraction tools process abstracts rather than full-text articles. The increasing availability of full text will allow more knowledge to be extracted with greater reliability. To investigate the challenges of full-text processing, we manually annotated a corpus of cited articles from a Molecular Interaction Map (Kohn, 1999).

Our analysis demonstrates the necessity of full-text processing; identifies the article sections where interactions are most commonly stated; and quantifies both the amount of external knowledge required and the proportion of interactions requiring multiple or deeper inference steps. Further, it identifies a range of NLP tools required, including: identifying synonyms, and resolving coreference and negated expressions. This is important guidance for researchers engineering biomedical text processing systems.

1 Introduction

It is no longer feasible for biologists to keep abreast of the vast quantity of biomedical literature. Even keyword-based Information Retrieval (IR) over abstracts retrieves too many articles to be individually inspected. There is considerable interest in NLP systems that overcome this information bottleneck.

Most bioNLP systems have been applied to abstracts only, due to their availability (Hirschman et al., 2002). Unfortunately, the information in abstracts is dense but limited. Full-text articles have the advantage of providing more information and repeating facts in different contexts, increasing the likelihood of an imperfect system identifying them.

Full text contains explicit structure, e.g. sections and captions, which can be exploited to improve Information Extraction (IE) (Regev et al., 2002). Previous work has investigated the importance of extracting information from specific sections, e.g. Schuemie et al. (2004), but there has been little analysis of when the entire document is needed for accurate knowledge extraction. For instance, extracting a fact from the Results may require a synonym to be resolved that is only mentioned in the Introduction. External domain knowledge may also be required.

We investigated these issues by manually annotating full-text passages that describe the functional relationships between bio-entities summarised in a *Molecular Interaction Map* (MIM). Our corpus tracks the process Kohn (1999) followed in summarising interactions for the mammalian cell MIM, by identifying information required to infer facts, which we call *dependencies*. We replicate the process of manual curation and demonstrate the necessity of full-text processing for fact extraction.

In the same annotation process we have identified NLP problems in these passages which must be solved to identify the facts correctly including: synonym and hyponym substitution, coreference resolution, negation handling, and the incorporation of knowledge from within the full text and the domain. This allows us to report on the relative importance of anaphora resolution and other tasks to the problem of biomedical fact extraction.

As well as serving as a dataset for future tool development, our corpus is an excellent case study providing valuable guidance to developers of biomedical text mining and retrieval systems.

BioNLP 2007: Biological, translational, and clinical language processing, pages 171–178,
Prague, June 2007.

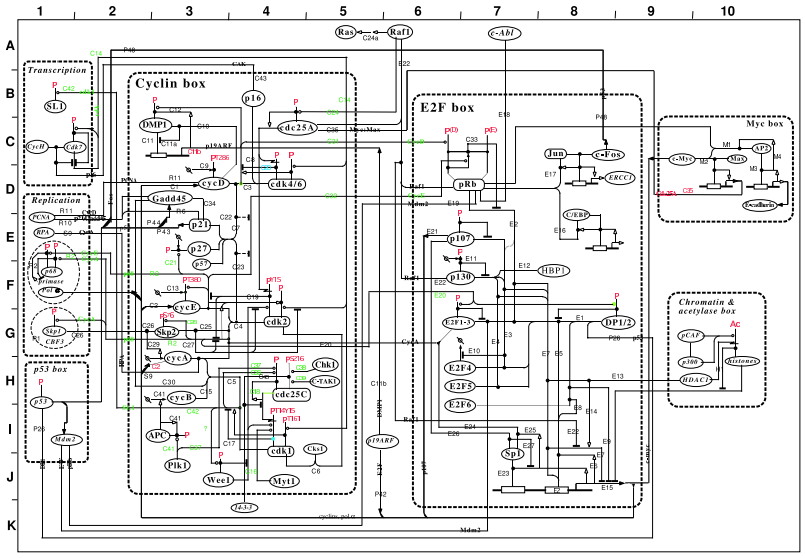


Figure 1: Map A of the Molecular Interaction Map compiled by Kohn (1999)

2 Biomedical NLP

Full-text articles are becoming increasingly available to NLP researchers, who have begun investigating how specific sections and structures can be mined in various information extraction tasks. Regev et al. (2002) developed the first bioIR system specifically focusing on limited text sections. Their performance in the KDD Cup Challenge, primarily using Figure legends, showed the importance of considering document structure. Yu et al. (2002) showed that the Introduction defines the majority of synonyms, while Schuemie et al. (2004) and Shah et al. (2003) showed that the Results and Methods are the most and least informative, respectively. In contrast, Sinclair and Webber (2004) found the Methods useful in assigning Gene Ontology codes to articles.

These section specific results highlight the information loss resulting from restricting searches to individual sections, as sections often provide unique information. Furthermore, facts appearing in different contexts across various sections, will be lost. This redundancy has been used for passage validation and ranking (Clarke et al., 2001).

There are limited training resources for biomedical full-text systems. The majority of corpora consist of abstracts annotated for bio-entity recognition and Relationship Extraction, such as the GENIA (Kim et al., 2003) and the BioCreAtIvE corpora. However, due to the lack of full-text corpora, many current systems only process abstracts (Ohta et al., 2006). Few biomedical corpora exist for other tasks, such as coreference resolution (Castaño et al., 2004; Vlachos et al., 2006), and these are very small. In this paper, we estimate the importance of these tasks in bioNLP systems, which will help determine which tasks system developers should focus effort on first.

Despite limited full-text training corpora, competitions such as the Genomics track of TREC, require systems to retrieve and rank passages from full text that are relevant to question style queries.

3 Molecular Interaction Maps

Kohn (1999) constructed a *Molecular Interaction Map* (MIM) based on literature describing 203 different interactions between bio-entities, such as proteins and genes, in mammalian cells (Figure 1). Interactions in the MIM are represented as links between nodes labelled with the bio-entities. Each link is associated with a description that summarises the evidence for the interaction from the literature, including citations. For example, Table 1 contains the description passage for interaction M4 (on the right of the Myc Box at grid reference C10 in Figure 1). Although MIM interactions may be mentioned in other articles, the articles cited by Kohn (1999) document the main biomedical research leading to the discovery of these interactions.

c-Myc and pRb enhance transcription from the E-cadherin promoter in an AP2-dependent manner in epithelial cells (mechanism unknown) (Batsche et al., 1998). Activation by pRb and c-Myc is not additive, suggesting that they act upon the same site, thereby perhaps blocking the binding of an unidentified inhibitor. No c-Myc recognition element is required for activation of the E-cadherin promoter by c-Myc. Max blocks transcriptional activation from the E-cadherin promoter by c-Myc, presumably because it blocks the binding between c-Myc and AP2.

Table 1: MIM annotation M4

1. M4 Subfact: *Activation of E-cadherin by pRb and c-Myc is not additive, suggesting they act on the same site*

a) However, the precise molecular mechanisms by which RB, Myc, and AP-2 cooperate to effect transcriptional activation of E-cadherin requires further study. . . . the positive effects of RB and c−Myc were not additive. (Discussion)
Synonym: *pRb equivalent to RB* – undefined
Synonym: *c-Myc equivalent to Myc*

b) The c-myc proto-oncogene, which encodes two amino-terminally distinct Myc proteins, acts as a transcription factor. (Intro)

Table 2: Example instances depending on synonym facts

In creating our corpus we have attempted to *reverse engineer* and document the MIM creation process for many of the interactions in Kohn (1999). We exhaustively traced and documented the process of identifying passages from the cited full-text articles that substantiate the MIM interactions. This allows us to identify and quantify the amount of information that is unavailable when systems are restricted to abstracts.

4 Corpus Creation

The first stage of corpus creation involved obtaining the full text of the articles cited in the MIM descriptions. There are 262 articles cited in Kohn (1999), and we have manually extracted the text from 218 of them; we have abstracts for the other 44 which have not been included in the analysis presented here.

Currently, the annotated part of the corpus consists of passages from 101 full-text articles, supporting 95 of the 203 MIM descriptions. A biomedical expert exhaustively identified these passages by manually reading each article several times. 30% of these articles support multiple MIM descriptions and so passages from these articles may appear multiple times. We restricted the corpus to the cited articles only. This allows us to quantify the need for external resources, e.g. synonym lists and ontologies. The corpus collection involved the following:

1. Each sentence in a MIM description is a called a *main fact*.
2. For each main fact we annotated every passage (*instance*) that the fact can be derived from. These include direct statements of the fact and passages the fact can be implied from.
3. Main facts are often complex sentences, combining numerous facts from the article. Passages from which part of a fact can be derived are also annotated as instances. A *subfact* is then created to represent these partial facts. This may be repeated for subfacts.
4. Many instances cannot be directly linked to their corresponding fact, as they *depend* on additional passages within the full text or external domain knowledge. New facts are formed to represent the dependency information – *synonym* and *extra* facts. Instances of these are annotated, and a link is added between the original and dependency facts.
5. Each instance is annotated with its location within the article. Linguistic phenomena, including anaphora, cataphora, and negated expressions which must be resolved to derive the fact are identified.

Tables 1 and 2 show an example of this process. One of the main facts of interaction M4 (Table 1) is *Activation by pRb and c-Myc is not additive . . . blocking the binding of an unidentified inhibitor.* An instance supporting part of this fact, the subfact in Table 2 *Activation of E-cadherin by pRb and c-Myc is not additive* . . . , 1.a), was identified. This instance requires the resolution of two synonymy dependencies, only one of which appears in the article.

2.	E13 Main Fact: *HDAC1 binds to the pocket proteins pRb, p107 and p130 and in turn is recruited to E2F complexes on promoters*
a)	The experiments described above indicate that p107 and p130 can interact with HDAC1. We thus reasoned that they could repress E2F activity by recruiting `histone deacetylase` activity to E2F containing promoters. (Results) Extra: *HDAC1 is a histone deacetylase*
b)	We have previously shown that Rb, the founding member of the pocket proteins family, represses E2F1 activity by recruiting the histone deacetylase HDAC1. (Abstract)

Table 3: Example instances depending on extra facts

3.	N4 Main fact: *RPA2 binds XPA via the C-terminal region of RPA2* Mutant RPA that lacked the p34 C terminus failed to interact with XPA, whereas RPA containing the p70 mutant (Delta RS) interacted with XPA (Fig. 2). (Results)
4.	C9 Subfact: *Cyclin D1 degraded rapidly by phosphorylation at threonine-286* Although "free" or CDK4-bound cyclin D1 molecules are intrinsically unstable (t1/2 $<$ 30 min), a cyclin D1 mutant (T286A) containing an alanine for threonine-286 substitution fails to undergo efficient polyubiquitination in an in vitro system or in vivo, and it is markedly stabilized (t1/2 approximately 3.5 hr) when inducibly expressed in either quiescent or proliferating mouse fibroblasts. (Abstract)

Table 4: Example instances with negated expressions

5 Dependencies

In our corpus, an instance of a fact may depend on additional facts (*dependencies*) to allow the fact to be derived from the original instance. Dependencies may occur elsewhere in the document or may not be mentioned at all. We consider two types of dependencies: synonym facts and extra facts.

5.1 Synonym Facts

The frequent use of synonyms, abbreviations and acronyms in biomedical text is a common source of ambiguity that is often hard to resolve (Sehgal et al., 2004). Furthermore, synonym lists are difficult to maintain in rapidly moving fields like biology (Lussier et al., 2006). There has been recent interest in developing systems to identify and extract these (Ao and Takagi, 2005; Okazaki and Ananiadou, 2006).

In our corpus we group all of these synonyms, abbreviations, acronyms and other orthographic variations as *synonym facts*. For example, the synonyms (1) `E2F4`, (2) `E2F-4` and (3) `E2F1-4` in our corpus refer to the same entity `E2F4`, however term (3) also includes the entities `E2F1`, `E2F2` and `E2F3`.

In Table 2, an instance supporting subfact 1. is shown in 1.a). The bio-entity `pRb` mentioned in the subfact does not appear in this instance. Thus 1.a) depends on knowing that `pRb` is equivalent to `RB`, and so we form a new synonym fact. This synonym is undefined in the article and cannot be assumed as `RB` is also a homograph for the gene ruby (`rb`), rubidium (`Rb`) and Robertsonian (`Rb`) translocations.

Instance 1 also depends on a second synonym – `c-Myc` and `Myc` are used interchangeably, where the protein `Myc` is referred to by its gene name, `c-Myc`. Metonymy is common in biology, and an instance supporting this synonym fact was found in the article, 1.b).

5.2 Extra Facts

Extra facts include all assertions (excluding synonym definitions) which are necessary to make a valid inference from an instance to a fact or subfact. These extra facts must be found within the same article. Many extra facts are descriptions or classes of bio-entities and hyponym relationships. According to Nédellec et al. (2006), a clearer distinction between entities and their classes/descriptions is needed in bioNLP corpora.

Example 2 in Table 3 is an instance which depends on an extra fact, 2.b), to derive the main fact. The class of proteins `histone deacetylase` in sentence 2 must be linked to the specific protein `HDAC1` in sentence 1, since the sortal anaphor `they` in sentence 2 refers to the antecedents `p107` and `p130`, and does not include `HDAC1`. This extra fact is identified in the apposition `the histone deacetylase HDAC1` in instance 2.b).

5.	C11b Subfact: *p19ARF induces cell cycle arrest in a p53-dependent manner* INK4a/ARF is perhaps the second most commonly disrupted locus in cancer cells. It encodes two distinct tumor suppressor proteins: p16INK4a, which inhibits the phosphorylation of the retinoblastoma protein by cyclin D-dependent kinases, and p19ARF, which stabilizes and activates p53 to promote either cell cycle arrest or apoptosis. (Intro)
6.	C36 Main fact: *Cdc25C is phosphorylated by Cyclin B-cdk1* In this work, we examine the effect of phosphorylation on the human cdc25-C protein (Sadhu et al.,1990). We show that this protein is phosphorylated during mitosis in human cells and that this requires active cdc2-cyclin B. (Intro)

Table 5: Example instances with cataphora and event anaphora

6 Negated Expressions

To quantify the importance of lexical and logical negations we have annotated each instance involving one or more negated expressions that must be resolved to derive the fact. In biomedical literature, negated expressions are commonly used to describe an abnormal condition, such as a mutation, and its resulting abnormal outcome, such as cancer, from which the normal condition and outcome can be inferred. This typically requires two or more negated expressions to be processed simultaneously.

Table 4 shows examples of instances with negated expressions. In the subject NP of instance 3, the lexical negative form of RPA (Mutant RPA) is followed directly by a logical negative detailing the function it failed to perform. These two negative expressions support the positive in the main fact. This implicit reporting of results expressed in terms of negative experimental outcomes is very common in molecular biology and genetics.

Example 4 requires external domain knowledge. Firstly, the amino acid alanine cannot be phosphorylated like threonine. Secondly, polyubiquitination triggers a signal for a protein (cyclin D1) to be degraded. Therefore from this negated pair the positive fact from interaction C9 can be inferred.

The context surrounding potential negative expressions must be analysed to determine if it is indeed a negative. For example, not all mutations result in negative outcomes – the mutation of p70 in instance 3 did not have a negative outcome.

7 Coreference Expressions

In biomedical literature, coreference expressions are used to make abbreviated or indirect references to bio-entities or events, and to provide additional information, such as more detailed descriptions.

To quantify the importance of coreference expressions, instances in our corpus are annotated with pronominal, sortal and event anaphoric, and cataphoric expressions, including those extending beyond one sentence. Instances 4–6 in Tables 4–5, each contain annotated pronominal or sortal anaphoric expressions. Instance 5 also involves a cataphoric expression, where suppressor proteins refers to p16INK4a and p19ARF

Event anaphora refer to processes and are quite common in biomedical text. We have annotated these separately to pronominal and sortal anaphora. Our event anaphora annotations are different to Humphreys et al. (1997). They associate sequential events, while we only refer to the same event.

An example is shown in instance 6 (Table 5) where the additional sortal anaphor complicates resolving the event anaphor. The third this refers to the phosphorylation event, phosphorylated, and not the protein cdc25-C like the second this.

8 Locating Facts

The key facts and results are generally repeated and reworded in various contexts within an article. This redundancy can be used in two ways to improve system precision and recall. Firstly, the redundancy increases the chance of an imperfect system identifying at least one instance. Secondly, the redundancy can be used for fact validation. By annotating every instance that supports a fact we are able to measure the degree of factual redundancy in full-text articles.

We have also annotated each instance with its location within the article: which section (or structure such as a title, heading or caption) it was contained within and the number of the paragraph. Using this data, we can evaluate the informativeness of each section and structure for identifying interactions.

Using our detailed dependency annotations we can also determine how many instances need addi-

Location	Main Fact	Subfact	Synonym	Extra
Title	3.3 (0.2)	1.9 (0.7)	0.0 (0.0)	0.8 (0.8)
Abstract	19.1 (10.1)	9.3 (5.1)	36.2 (21.7)	25.8 (14.8)
Introduction	11.3 (5.2)	8.3 (3.4)	30.4 (17.4)	17.2 (7.8)
Results	31.0 (13.8)	37.6 (16.1)	20.3 (15.9)	32.0 (12.5)
Discussion	21.8 (7.3)	19.5 (6.6)	2.9 (1.4)	9.4 (3.1)
Figure Heading	5.0 (0.6)	10.7 (3.8)	1.4 (1.4)	2.3 (0.0)
Figure Legend	3.1 (1.3)	4.8 (2.0)	0.0 (0.0)	7.0 (4.7)
Table Data	0.0 (0.0)	0.2 (0.0)	0.0 (0.0)	0.0 (0.0)
Methods	0.2 (0.0)	0.1 (0.1)	0.0 (0.0)	4.7 (0.8)
Conclusion	0.6 (0.4)	0.1 (0.0)	0.0 (0.0)	0.0 (0.0)
Footnotes	0.0 (0.0)	0.0 (0.0)	5.8 (2.9)	0.0 (0.0)
Headings	4.8 (0.6)	7.5 (2.7)	2.9 (1.4)	0.8 (0.8)
Full-text	100.0 (39.4)	100.0 (40.6)	100.0 (62.3)	100.0 (45.3)

Table 6: Instances found excluding (including) all dependencies

Fact Type	# Created	# Found	# Instances
Main Fact	170	156	523
Subfact	251	251	1196
Synonym	155	62	69
Extra	152	87	128
Total	728	556	1916

Table 7: Distribution of fact types in corpus

Instances	Synonym	Extra
Main Fact	46.8 (10.9)	26.2 (18.9)
Subfact	36.9 (8.2)	26.7 (15.4)
Synonym	8.7 (2.9)	7.2 (4.3)
Extra	25.0 (0.0)	13.3 (10.9)

Table 8: Instances with (all found) dependencies

tional knowledge outside of the current section to support a particular fact. This demonstrates how important full-text processing is.

9 Corpus Analysis

Having described the corpus annotation we can now investigate various statistical properties of the data. Table 7 shows the distribution of the various annotated fact types within the corpus. There are a total of 728 different facts identified, with 556 (76%) found within the documents. We have annotated 1916 individual passages as instances, totally 2429 sentences. There were 14 main facts that we found no instances or subfact instances for.

The most redundancy occurs in main facts and subfacts, with on average 3.35 and 4.76 instances each respectively, whilst synonym facts have almost no redundancy. Also, a large proportion of synonym and extra facts, 60% and 43% respectively, do not appear anywhere in the articles (Table 7).

This high level of redundancy in facts demonstrates the significant advantages of processing full text. However, the proportion of missing synonym and extra facts shows the importance of external resources, such as synonym lists, and tools for recognising orthographic variants.

9.1 Locating Facts

Table 6 shows the percentage of instances identified in particular locations within the articles. The best sections for finding instances of facts and subfacts were the Results and Discussion sections, whereas synonym and extra facts were best found in the Abstract, Introduction and Results. The later sections of each article rarely contributed any instances. Interestingly, we did not find the Figure headings or legends to be that informative for main facts. Figure headings are restricted in length and thus are rarely able to express main facts as well as subfacts.

The proportion of main facts and subfact instances found in the abstract is quite small, further demonstrating the value of full-text processing.

If we take into account the additional dependency information, and restrict the instances to those fully supported within a given section, the results drop dramatically (those in parentheses in Table 6). In

Depth	Fact	Subfact	Synonym	Extra
0	35.2	45.1	87.0	64.8
1	53.9	44.2	13.0	26.6
2	9.6	9.5	0.0	7.0
3	1.3	0.9	0.0	1.6
4	0.0	0.3	0.0	0.0

Table 9: Maximum depth of instance dependencies

Breadth	Fact	Subfact	Synonym	Extra
0	35.2	45.1	87.0	64.8
1	36.5	35.5	7.2	29.7
2	22.6	15.7	5.8	4.7
3	4.6	2.9	0.0	0.8
4	0.8	0.6	0.0	0.0
5	0.2	0.2	0.0	0.0

Table 10: Breadth of instance dependencies

total, the number of instances drops to 39.4% and 40.6%, for main facts and subfacts, respectively. This again demonstrates the need for full-text processing, including the dependencies between facts found in different sections of the article.

9.2 Dependencies

Our corpus represents each of the facts and subfacts as a dependency graph of instances, each which in turn may require support from other facts, including synonym and extra facts.

Table 8 shows the percentage of instances which depend on synonym and extra facts in our corpus. 46.8% of main fact instances depend on at least one synonym fact, but only 10.9% of main fact instances which depend on at least one synonym were completely resolved (i.e. all of the synonyms were found as well). Interestingly, synonym and extra facts often required other synonym and extra facts.

Our corpus contains more synonym than extra fact dependencies, however more extra facts were defined in the articles. The large proportion of main facts and subfacts depending on synonyms and extra facts demonstrates the importance of automatically extracting this information from full text.

Since the inference from an instance to a fact may depend on other facts, long chains of dependencies may occur, all of which would need to be resolved before a main fact could be derived from the text.

Expressions	Instances
Negated	4.3
Anaphora	13.2
Event Anaphora	6.6
Cataphora	2.7

Table 11: Distribution of annotated expressions

Table 9 shows the distribution of maximum chain depth in our dependency graphs. The maximum depth is predominately less than 3. Table 10 shows the distribution of the breadth of dependency graphs. Again, most instances are supported by fewer than 3 dependency chains. Most instances depend on some other information, but luckily, a large proportion of those only require information from a small number of other facts. However, given that these facts could occur anywhere within the full text, extracting them is still a very challenging task.

9.3 Negated & Coreference Expressions

Table 11 shows the percentage of instances annotated with negated, anaphoric and cataphoric expressions in our corpus. We have separated event anaphora from pronominal and sortal anaphora. There are fewer cataphoric and negated expressions than anaphoric expressions. Therefore, we would expect the greatest improvement when systems incorporate anaphora resolution components, and little improvement from cataphoric and negated expression analysis. However, negated expressions provide valuable information regarding experimental conditions and outcomes, and thus may be appropriate for specific extraction tasks.

10 Conclusion

This paper describes a corpus documenting the manual identification of facts from full-text articles by biomedical researchers. The corpus consists of articles cited in a Molecular Interaction Map developed by Kohn (1999). Each fact can be derived from one or more passages from the citations. Each of these *instances* was annotated with their location in the article and whether they contained coreference or negated expressions. Each instance was also linked with other information, including synonyms and extra knowledge, that was required to derive the particular interaction. The annotation task was quite com-

plex and as future work we will increase the reliability of our corpus by including the annotations of other domain experts using our guidelines, and use this resource for tool development. The guidelines and corpus will be made publicly available.

Our corpus analysis demonstrates that full-text analysis is crucial for exploiting biomedical literature. Less than 20% of fact instances we identified were contained in the abstract. Analysing sections in isolation reduced the number of supported facts by 60%. We also showed that many instances were dependent on a significant amount of other information, both within and outside the article. Finally, we showed the potential impact of various NLP components such as anaphora resolution systems.

This work provides important empirical guidance for developers of biomedical text mining systems.

Acknowledgements

This work was supported by the CSIRO ICT Centre and ARC Discovery grants DP0453131 and DP0665973.

References

Hiroko Ao and Toshihisa Takagi. 2005. ALICE: An algorithm to extract abbreviations from Medline. *Journal of the American Medical Informatics Association*, 12(5):576–586.

J. Castaño, J. Zhang, and J. Pustejovsky. 2004. Anaphora resolution in biomedical literature. In *International Symposium on Reference Resolution in NLP*, Alicante, Spain.

Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. 2001. Exploiting redundancy in question answering. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 358–365, New Orleans, LA.

Lynette Hirschman, Jong C. Park, Junichi Tsujii, Limsoon Wong, and Cathy Wu. 2002. Accomplishments and challenges in literature data mining for biology. *Bioinformatics Review*, (12):1553–1561.

Kevin Humphreys, Robert Gaizauskas, and Saliha Azzam. 1997. Event coreference for information extraction. In *Proc. of the ACL/EACL Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, Madrid, Spain.

Jin-Dong Kim, Tomoko Ohta, Yuka Teteisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(1):i180–i182.

Kurt W. Kohn. 1999. Molecular interaction map of the mammalian cell cycle and DNA repair systems. *Molecular Biology of the Cell*, 10:2703–2734.

Yves Lussier, Tara Borlawsky, Daniel Rappaport, Yang Liu, and Carol Friedman. 2006. PHENOGO: Assigning phenotypic context to gene ontology annotations with natural language processing. In *Proc. of the Pacific Symposium on Biocomputing*, volume 11, pages 64–75, Maui, HI.

Clair Nédellec, Philippe Bessières, Robert Bossy, Alain Kptoujanksy, and Alain-Pierre Manine. 2006. Annotation guidelines for machine learning-based named entity recognition in microbiology. In *Proc. of the ACL Workshop on Data and Text for Mining Integrative Biology*, pages 40–54, Berlin.

Tomoko Ohta, Yusuke Miyao, Takashi Ninomiya, Yoshimasa Tsuruoka, Akane Yakushiji, Katsuya Masuda, Jumpei Takeuchi, Kazuhiro Yoshida, Tadayoshi Hara, Jin-Dong Kim, Yuka Tateisi, and Jun'ichi Tsujii. 2006. An intelligent search engine and GUI-based efficient Medline search tool based on deep syntactic parsing. In *Proc. of the COLING/ACL Interactive Presentation Sessions*, pages 17–20, Sydney, Australia.

Naoaki Okazaki and Sophia Ananiadou. 2006. A term recognition approach to acronym recognition. In *Proc. the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*, pages 643–650, Sydney, Australia.

Yizhar Regev, Michal Finkelstein-Langau, Ronen Feldman, Mayo Gorodetsky, Xin Zheng, Samuel Levy, Rosane Charlab, Charles Lawrence, Ross A. Lippert, Qing Zhang, and Hagit Shatkay. 2002. Rule-based extraction of experimental evidence in the biomedical domain - the KDD Cup 2002 (Task 1). *ACM SIGKKD Explorations*, 4(2):90–92.

M.J. Schuemie, M. Weeber, B.J.A. Schijvenaars, E.M. van Mulligen, C.C. van der Eijk, R.Jelier, B.Mons, and J.A Kors. 2004. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, 20(16):2597–2604.

Aditya K. Sehgal, Padmini Srinivasan, and Olivier Bodenreider. 2004. Gene terms and english words: An ambiguous mix. In *Proc. of the ACM SIGIR Workshop on Search and Discovery for Bioinformatics*, Sheffield, UK.

Parantu K. Shah, Carolina Perez-Iratxeta, Peer Bork, and Miguel A. Andrade. 2003. Information extraction from full text scientific articles: where are the keywords? *BMC Bioinformatics*, 4(20).

Gail Sinclair and Bonnie Webber. 2004. Classification from full text: A comparison of canonical sections of scientific papers. In *Proc. of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 66–69, Geneva, Switzerland.

Andreas Vlachos, Caroline Gasperin, Ian Lewin, and Ted Briscoe. 2006. Bootstrapping the recognition and anaphoric linking of named entities in drosophila articles. In *Proc. of the Pacific Symposium on Biocomputing*, volume 11, pages 100–111, Maui, HI.

Hong Yu, Vasileios Hatzivassiloglou, Carol Friedman, Andrey Rzhetsky, and W.John Wilbur. 2002. Automatic extraction of gene and protein synonyms from Medline and journal articles. In *Proc. of the AMIA Symposium 2002*, pages 919–923, San Antonio, TX.

Adaptation of POS Tagging for Multiple BioMedical Domains

John E. Miller[1] **Manabu Torii**[2] **K. Vijay-Shanker**[1]

[1]Computer & Information Sciences
University of Delaware
Newark, DE 19716
{jmiller,vijay}@cis.udel.edu

[2]Biostatistics, Bioinformatics and Biomathematics
Georgetown University Medical Center
Washington, DC 20057
mt352@georgetown.edu

1 Introduction

Part of Speech (POS) tagging is often a prerequisite for tasks such as partial parsing and information extraction. However, when a POS tagger is simply ported to another domain the tagger's accuracy drops. This problem can be addressed through hand annotation of a corpus in the new domain and supervised training of a new tagger. In our methodology, we use existing raw text and a generic POS annotated corpus to develop taggers for new domains without hand annotation or supervised training. We focus in particular on out-of-vocabulary words since they reduce accuracy (Lease and Charniak. 2005; Smith et al. 2005).

There is substantial information in the derivational suffixes and few inflectional suffixes of English. We look at individual words and their suffixes along with the morphologically related words to build a domain specific lexicon containing POS tags and probabilities for each word.

2 Adaptation Methodology

Our methodology is described in detail in Miller et al (2007) and summarized here: 1) Process generic POS annotated text to obtain state and lexical POS tag probabilities. 2) Obtain a frequency table of words from a large corpus of raw sub-domain text. 3) Construct a partial sub-domain lexicon matching relative frequencies of morphologically related words with words from the generic annotated text averaging POS probabilities of the k nearest neighbors. 4) Combine common generic words and orthographic word categories with the partial lexicon making the sub-domain lexicon. 5) Train a first order Hidden Markov Model (HMM) by Expectation Maximization (EM). 6) Apply the Viterbi algorithm with the HMM to tag sub-domain text.

3 Adaptation to Multiple Domains

Molecular Biology Domain: We used the Wall Street Journal corpus (WSJ) (Marcus et al, 1993) as our generic POS annotated corpus. For our raw un-annotated text we used 133,666 abstracts from the MEDLINE distribution covering molecular biology and biomedicine sub-domains. We split the GENIA database (Tateisi et al, 2003) into training and test portions and ignored the POS tags for training. We ran a 5-fold cross validation study and obtained an average accuracy of 95.77%.

Medical Domain: Again we used the WSJ as our generic POS annotated corpus. For our raw un-annotated text we used 164,670 abstracts from the MEDLINE distribution with selection based on 83 journals from the medical domain. For our HMM EM training we selected 1966 abstracts (same journals). For evaluation purposes, we selected 1932 POS annotated sentences from the MedPost (Smith et al, 2004) distribution (same journals). The MedPost tag set coding was converted to the Penn Treebank tag set using the utilities provided with the MedPost tagger distribution. We obtained an accuracy of 93.17% on the single medical test corpus, a substantial drop from the 95.77% average accuracy obtained in the GENIA corpus.

4 Coding Differences

We looked at high frequency tagging errors in the medical test set and found that many errors resulted directly from the differences in the coding styles between GENIA and MedPost. Our model reflects the coding style of the WSJ, used for our generic POS annotated text. GENIA largely followed the WSJ coding conventions. Annotation in the 1932 sentences taken from MedPost had some systematic differences in coding style from this.

BioNLP 2007: Biological, translational, and clinical language processing, pages 179–180,
Prague, June 2007.

Identified Differences: Lexical differences: 1) Words such as 'more' and 'less' are JJR or RBR in WSJ/GENIA but JJ or RB in MedPost. 2) Tokens such as %, =, /, <, > are typically NN or JJ in WSJ/GENIA but SYM in MedPost. 3)'be' is VB in WSJ/GENIA but VB or VBP in MedPost. 4) Some orthographic categories are JJ in WSJ/GENIA but NN in MedPost. Transition discrepancies: 1) Verbs are tagged VB following a TO or MD in WSJ/GENIA but only following a TO in MedPost. 2) MedPost prefers NN and NN-NN sequences.

***Ad Hoc* Adjustments**: We constructed a new lexicon accounting for some of the lexical differences and attained an accuracy of 94.15% versus the previous 93.17%. Next we biased a few initial state transition probabilities, changing P(VB|MD) from very high to a very low and increasing P(NN|NN), and attained an accuracy of 94.63%.

As the coding differences had nothing to do with suffixes and suffix distributions, the central part of our methodology, we tried some *ad hoc* fixes to determine what our performance might have been. We suffered at least a 1.46% drop in accuracy due to differences in coding, not language use.

5 Evaluation

The table shows the accuracy of our tagger and a few well-known taggers in our target biomedical sub-domains.

Molecular Biology	%Accuracy
- Our tagger (5-fold)	95.8%
- MedPost	94.1%
- Penn BioIE[1]	95.1%
- GENIA supervised	98.3%
Medical Domain	
- Our tagger	93.17%
- Our tagger (+ lex bias)	94.15%
- Our tagger (+ lex & trans bias)	94.63%
- MedPost supervised[2]	96.9%

The MedPost and Penn BioIE taggers used annotated text and supervised training in other biomedical domains, but they were not trained specifically for the GENIA Molecular Biology subdomain. Our tagger seems competitive with these taggers. We cannot claim superior accuracy as these taggers may suffer the same coding bias effects we have noted. The superior performance of the GENIA tagger (Tsuruoka et al. 2005) in the Molecular Biology/GENIA domain and the MedPost tagger (Smith et al. 2004) in its biomedical domain owes to their use of supervised training on an annotated training set with evaluation on a test set from the same domain. The approximate 1.5% bias effect due to coding differences is attributable to organizational differences in POS.

[1] PennBioIE. 2005. Mining The Bibliome Project. http://bioie.ldc.upenn.edu/.

[2] Based on Medpost test set of 1000 sentences, not on our test set of 1932 sentences.

6 Conclusions

To cope with domain specific vocabulary and uses of vocabulary, we exploited the suffix information of words and related words to build domain specific lexicons. We trained our HMM using EM and un-annotated text from the specialized domains. We assessed accuracy versus annotated test sets in the specialized domains, noting discrepancies in our results across specialized domains, and concluding that our methodology performs competitively versus well-known taggers that used annotated text and supervised training in other biomedical domains.

References

M. Lease and E. Charniak. 2005. Parsing Biomedical Literature. IJCNLP-05: 58-69.

M. Marcus, B. Santorini, M.A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. Comp. Ling., 19:313-330.

J.E. Miller, M. Torii, K. Vijay-Shanker. 2007. Building Domain-Specific Taggers Without Annotated (Domain) Data. EMNLP-07.

L. Smith, T. Rindflesch, W.J. Wilbur. 2004. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics 20 (14):2320-2321.

L. Smith, T. Rindflesch, W.J. Wilbur. 2005. The importance of the lexicon in tagging biomedical text. Natural Language Engineering 12(2) 1-17.

Y. Tateisi, T. Ohta, J. Dong Kim, H. Hong, S. Jian, J. Tsujii. 2003. The GENIA corpus: Medline abstracts annotated with linguistic information. Third meeting of SIG on Text Mining, ISMB.

Y. Tsuruoka, Y. Tateishi, J.D. Kim, T. Ohta, J. McNaught, S. Ananiadou, J. Tsujii. 2005. Developing a Robust Part-of-Speech Tagger for Biomedical Text, Advances in Informatics, LNCS 3746: 382-392.

Information Extraction from Patients' Free Form Documentation

Agnieszka Mykowiecka
Institute of Computer Science, PAS
Ordona 21, 01-237 Warszawa, Poland
agn@ipipan.waw.pl

Małgorzata Marciniak
Institute of Computer Science, PAS
Ordona 21, 01-237 Warszawa, Poland
mm@ipipan.waw.pl

Abstract

The paper presents two rule-based information extraction (IE) from two types of patients' documentation in Polish. For both document types, values of sets of attributes were assigned using specially designed grammars.

1 Method/General Assumptions

Various rule-based, statistical, and machine learning methods have been developed for the purpose of information extraction. Unfortunately, they have rarely been tested on Polish texts, whose rich inflectional morphology and relatively free word order is challenging. Here, we present results of two experiments aimed at extracting information from mammography reports and hospital records of diabetic patients.[1] Since there are no annotated corpora of Polish medical text which can be used in supervised statistical methods, and we do not have enough data for weakly supervised methods, we chose the rule-based extraction schema. The processing procedure in both experiments consisted of four stages: text preprocessing, application of IE rules based on the morphological information and domain lexicons, postprocessing (data cleaning and structuring), and conversion into a relational database.

Preprocessing included format unification, data anonymization, and (for mammography reports) automatic spelling correction.

The extraction rules were defined as grammars of the SProUT system, (Drożdżyński et al., 2004). SProUT consists of a set of processing components for basic linguistic operations, including tokenization, sentence splitting, morphological analysis (for Polish we use Morfeusz (Woliński, 2006)) and gazetteer lookup. The SproUT components are combined into a pipeline that generates typed feature structures (TFS), on which rules in the form of regular expressions with unification can operate. Small specialized lexicons containing both morphological and semantic (concept names) information have been created for both document types.

Extracted attribute values are stored in a relational database.[2] Before that, mammography reports results undergo additional postprocessing — grouping together of extracted data. Specially designed scripts put limits that separate descriptions of anatomical changes, tissue structure, and diagnosis. More details about mammography IE system can be found in (Mykowiecka et al., 2005).

2 Document types

For both document types, partial ontologies were defined on the basis of sample data and expert knowledge. To formalize them, we used OWL-DL standard and the *Protégé* ontology editor. The excerpt from the ontology is presented in Fig. 1.

In both cases, the relevant part of the ontology was translated into a TFS hierarchy. This resulted in 176 types with 66 attributes for the mammography domain, and 139 types (including 75 drug names) with 65 attributes for diabetic patients' records.

[1]This work was partially financed by the Polish national project number 3 T11C 007 27.

[2]This last stage is completed for the diabetes reports while for mammography it is still under development.

BioNLP 2007: Biological, translational, and clinical language processing, pages 181–182,
Prague, June 2007.

BiochemicalData: BloodData: HB1C
Diet
DiseaseOrSymptom
  Disease
    AutoimmuneDisease
    Cancer
    Diabetes: Type1, Type2, TypeOther
  Symptom
    Angiopathy: Macroangiopathy, Microangiopathy
    BoodSymptom: Hypoglicaemia
    Neuropathy: Autonomic, PeripheralPolineuropathy
    UrineSymptom: Acetonuria, Microalbuminuria
Medicine
  DiabeticMedicine: Insulin, OralDiabeticMedicine
AnatomicalLocalization
  BodyPart
    Breast: Subareola, urq, ulq, lrq, llq
  BodySide: Left, Right
HistDiagnosis: Benign, Suspicious, Malignant
TissueSpecification: GlandularTissue, FatTissue

Figure 1: A sample of classes

3 Extraction Grammars

The number of rules is highly related to the number of attributes and possible ways of formulating their values. The grammar for mammography reports contains 190 rules; that for hospital records contains about 100 rules. For the first task, nearly the entire text is covered by the rules, while for the second, only a small part of the text is extracted (e.g., from many blood tests we are interested only in HBA1C). Polish inflection is handled by using the morphological analyzer and by inserting the most frequent morphological forms into the gazetteer. Free word order is handled either by rules which describe all possible orderings, or by extracting small pieces of information which are merged at the postprocessing stage. Fig. 2 presents a fragment of one mammography note and its output. The *zp* and *zk* markers are inserted during the information structuring stage to represent borders of an anatomical change description. Similar markers are introduced to structure the tissue description part.

4 Evaluation

The experiments were evaluated on a set of previously unseen reports. Extraction of the following structures was evaluated: 1) simple attributes (e.g. diabetes balance); 2) structured attributes (e.g. localization); and 3) complex structures (e.g. description of abnormal findings). Evaluation of three selected attributes from both sets is given in Fig. 3.

W obu sutkach rozsiane pojedyncze mikrozwapnienia o charakterze łagodnym. Doły pachowe prawidłowe. Kontrolna mammografia za rok.
(*Within both breasts there are singular benign microcalcifications. Armpits normal. Next control mammography in a year.*)

zp LOC|BODY_PART:breast||LOC|L_R:left-right
ANAT_CHANGE:micro||GRAM_MULT:plural
zk DIAGNOSIS_RTG:benign
DIAGNOSIS_RTG:no_susp||LOC_D|BODY_PART:
armpit||LOC_D|L_R:left-right
RECOMMENDATION|FIRST:mmg||TIME:year

Figure 2: A fragment of an annotated mammography report

The worse results for unbalanced diabetes recognition were due to an unpredicted expression type.

mammography – 705 reports			
	cases	precision	recall
findings	343	90.76	97.38
block beginnings	299	81.25	97.07
localizations	2189	98.42	99.59
diabetes – 99 reports			
unbalanced diabetes	58	96,67	69,05
diabetic education	39	97,50	97,50
neuropathy	30	100	96,77

Figure 3: Evaluation results for selected attributes

5 Conclusions

Despite the fact that rule based extraction is typically seen as too time consuming, we claim that in the case of very detailed information searching, designing rules on the basis of expert knowledge is in fact a method of a real practical value. In the next stage, we plan to use our tools for creating annotated corpora of medical texts (manually corrected). These data can be used to train statistical IE models and to evaluate other extraction systems.

References

Agnieszka Mykowiecka, Anna Kupść, Małgorzata Marciniak. 2005. Rule-based Medical Content Extraction and Classification, *Proc. of IIS: IIPWM05*. Advances in Soft Comp., Vol. 31, Springer-Verlag.

Witold Drożdżyński and Hans-Ulrich Krieger and Jakub Piskorski and Ulrich Schäfer and Feiyu Xu. 2004. Shallow Processing with Unification and Typed Feature Structures – Foundations and Applications. *German AI Journal KI-Zeitschrift, 01/04*.

Marcin Woliński. 2006. Morfeusz – a Practical Tool for the Morphological Analysis of Polish, *Proc. of IIS: IIPWM06*. Adv. in Soft Comp., Springer-Verlag.

Automatic Indexing of Specialized Documents: Using Generic *vs.* Domain-Specific Document Representations

Aurélie Névéol **James G. Mork** **Alan R. Aronson**
{neveola,mork,alan}@nlm.nih.gov
National Library of Medicine
8600 Rockville Pike
Bethesda, MD 20894
USA

Abstract

The shift from paper to electronic documents has caused the curation of information sources in large electronic databases to become more generalized. In the biomedical domain, continuing efforts aim at refining indexing tools to assist with the update and maintenance of databases such as MEDLINE®. In this paper, we evaluate two statistical methods of producing MeSH® indexing recommendations for the genetics literature, including recommendations involving subheadings, which is a novel application for the methods. We show that a generic representation of the documents yields both better precision and recall. We also find that a domain-specific representation of the documents can contribute to enhancing recall.

1 Introduction

There are two major approaches for the automatic indexing of text documents: statistical approaches that rely on various word counting techniques [such as vector space models (Salton, 1989), Latent Semantic Indexing (Deerwester et al., 1990) or probabilistic models (Sparck-Jones et al., 2000)] and linguistic approaches that involve syntactical and lexical analysis [see for example term extraction and term variation recognition in systems such as MetaMap (Aronson, 2001), FASTR (Jacquemin and Tzoukermann, 1999) or IndDoc (Nazarenko and Ait El Mekki, 2005)]. In many cases, the combination of these approaches has been shown to improve the performance of a single approach both for controlled indexing (Aronson et al., 2004) and free text indexing (Byrne and Klein, 2003).

Recently, Névéol et al. (2007) presented linguistic approaches for the indexing of documents in the field of genetics. In this paper, we explore a statistical approach of indexing for text documents also in the field of genetics. This approach was previously used successfully to produce Medical Subject Headings (MeSH) main heading recommendations. Our goal in this experiment is twofold: first, extending an existing method to the production of recommendations involving subheadings and second, assessing the possible benefit of using a domain-specific variant of the method.

2 A k-Nearest-Neighbors approach for indexing

2.1 Principle

The k-Nearest-Neighbors (k-NN) approach views indexing as a multi-class classification problem where a document may be assigned several "classes" in the form of indexing terms. It requires a large set of labeled data composed of previously indexed documents. k-NN relies on the assumption that similar documents should be classified in a similar way. The algorithm consists of two steps: 1/documents that are most "similar" to the query document must be retrieved from the set of labeled documents. They are considered as "neighbors" for the query document; 2/an indexing set must be produced from these and assigned to the query document.

Finding similar documents

All documents are represented using a vector of distinctive features within the representation space. Based on this representation, labeled documents

BioNLP 2007: Biological, translational, and clinical language processing, pages 183–190,
Prague, June 2007.

may be ranked according to their similarity to the query document using usual similarity measures such as cosine or Dice. The challenge in this step is to define an appropriate representation space for the documents and to select optimal features for each document. Another issue is the number (k) of neighbors that should be selected to use in the next step.

Producing an indexing set

When applied to a single-class classification problem, the class that is the most frequent among the k neighbors is usually assigned to the query document. Indexing is a multi-class problem for which the number of classes a document should be assigned is not known, as it may vary from one document to another. Therefore, indexing terms from the neighbor documents are all taken into account and ranked according to the number of neighbors that were labeled with them. The more neighbors labeled with a given indexing term, the higher the confidence that it will be a relevant indexing term for the query document. This resulting indexing set may then be filtered to select only the terms that were obtained from a defined minimum number of neighbors.

2.2 Document representation

Generic representation

A generic representation of documents is obtained from the text formed by the title and abstract. This text is processed so that punctuation is removed, stop-words from a pre-defined list (of 310 words) are removed, remaining words are switched to lower case and a minimal amount of stemming is applied. As described by Salton (1989) words should be weighted according to the number of times they occur in the query document and the number of times they occur in the whole collection (here, MEDLINE). Moreover, words from the title are given an additional weight compared to words from the abstract. Further adjustments relative to document length and local weighting according to the Poisson distribution are detailed in (Aronson et al, 2000; Kim et al., 2001) where the PubMed Related Citations (PRC) algorithm is discussed. Further experiments showed that the best results were obtained by using the ten nearest neighbors.

Domain-specific representation

In specialized domains, documents from the literature may be represented with concepts or objects commonly used or studied in the field. For example, (Rhodes et al., 2007) meet specific chemistry oriented search needs by representing US patents and patent applications with molecular information in the form of chemical terms and structures. A similar representation is used for PubChem (http://pubchem.ncbi.nlm.nih.gov/) records. In the genetics domain, genes are among the most commonly discussed or manipulated concepts. Therefore, genes should provide a relevant domain-specific description of documents from the genetics literature.

The second indexing algorithm that we describe in this paper, know as the Gene Reference Into Function (GeneRIF) Related Citations (GRC) algorithm, uses "GeneRIF" links (defined in the paragraph below) to retrieve neighbors for a query document.

To form a specific representation of the document, gene names are retrieved by ABGene[1] (Tanabe and Wilbur, 2002) and mapped to Entrez Gene[2] unique identifiers. The mapping was performed with a version of SemRep (Rindflesch and Fiszman, 2003) restricted to human genes. It consists in normalizing the gene name (switch to lower case, remove spaces and hyphens) and matching the resulting string to one of the gene names or aliases listed in Entrez Gene.

For each gene, the GeneRIF links supply a subset of MEDLINE citations manually selected by NLM indexers for describing the functions associated with the gene. These sets were used in two ways:

To complete the document representation. If a citation was included in the GeneRIF of a given gene, the gene was given an additional weight in the document representation.

To limit the set of possible neighbors. In the generic representation, all MEDLINE citations contain the representation features, words. Therefore, they all have to be considered as potential neighbors. However,

[1] Software downloaded January 17, 2007, from http://www.ncbi.nlm.nih.gov/staff/lsmith/MedPost.html

[2] Retrieved January 17, 2007, from: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene

only a subset of citations actually contains genes. Therefore, only those citations need to be considered as potential neighbors. This observation enables us to limit the specific processing to relevant citations. Possible neighbors for a query document consist of the union of the GeneRIF citations corresponding to each gene in the document representation.

Table 1: Gene description of a sample MEDLINE document and its two nearest neighbors

PubMed IDs	ABGene	Entrez Gene IDs
15645653	abcc6 mrp6 ldl-r pxe fh	368 368; 6283 3949 368; 5823 2271
10835643	mrp6 pxe	368; 6283 368; 5823
16392638	abcc6 mrp6 pxe	368 368; 6283 368; 5823

For each query document, the set of possible neighbors was processed and ranked according to gene similarity using a cosine measure. Table 1 shows the description of a sample MEDLINE citation and its two nearest neighbors.

Based on experiments with the PubMed Related Citations algorithm, ten neighbors were retained to form a candidate set of indexing terms.

3 Experiment

3.1 Application to MeSH indexing

In the MEDLINE database, publications of the biomedical domain are indexed with Medical Subject Headings, or MeSH descriptors. MeSH contains about 24,000 main headings denoting medical concepts such as *foot*, *bone neoplasm* or *appendectomy*. MeSH also contains 83 subheadings such as *genetics*, *metabolism* or *surgery* that can be associated with the main headings in order to refer to a specific aspect of the concept. Moreover, each descriptor (a main heading alone or associated with one or more subheadings) is assigned a "minor" or "major" weight depending on how substantially the concept it denotes is discussed in the article. "Major" descriptors are marked with a star.

In order to form a candidate indexing set to be assigned to a query document, the descriptors assigned to each of the neighbors were broken down into a set of main headings and pairs (i.e. a main heading associated with a single subheading). For this experiment, indications of major terms were ignored.

For example, the MeSH descriptor *Myocardium/cytology/metabolism would generate the main heading Myocardium and the two pairs Myocardium/cytology and Myocardium/metabolism.

3.2 Test Corpus

Both methods were tested on a corpus composed of a selection of the 49,863 citations entered into MEDLINE in January 2005. The 2006 version of MeSH was used for the indexing in these citations. About one fifth of the citations (10,161) are considered to be genetics-related, as determined by Journal Descriptor Indexing (Humphrey, 1999). Our test corpus was composed of genetics-related citations from which Entrez Gene IDs could be extracted – about 40% of the cases. The final test corpus size was 3,962. Appendix A shows a sample citation from the corpus.

3.3 Protocol

Figure 1 shows the setting of our experiment. Documents from the test corpus described above were processed to obtain both a generic and specific representation as described in section 2.2. The corresponding ten nearest neighbors were retrieved using the PRC and GRC algorithms. All the neighbors' MeSH descriptors were pooled to form candidate indexing sets of descriptors that were evaluated using precision and recall measures. Precision was the number of candidate descriptors that were selected as indexing terms by NLM indexers (according to reference MEDLINE indexing) over the total number of candidate descriptors. Recall was the number of candidate descriptors that were selected as indexing terms by NLM indexers over the total number of indexing terms expected (according to reference MEDLINE indexing). For better comparison between the methods, we also computed F-measure giving equal weight to preci-

sion and recall - F1=2*PR/(P+R) and giving a higher weight to recall - F3=10*PR/(9P+R).

Four different categories of descriptors were considered in the evaluation:

MH: MeSH main headings (regardless of whether subheadings were attached in the reference indexing)

SH: stand-alone subheadings (regardless of the main heading(s) they were attached to in the reference indexing)

MH/SH: main heading/subheading pairs

DESC: MeSH descriptors, i.e. main headings and main heading/subheading pairs

Similarly, four different candidate indexing sets were considered: the indexing set resulting from PRC, the indexing set resulting from GRC, the indexing set resulting from the pooling of PRC and GRC sets and finally the indexing set resulting from the intersection of PRC and GRC indexing sets (common index terms).

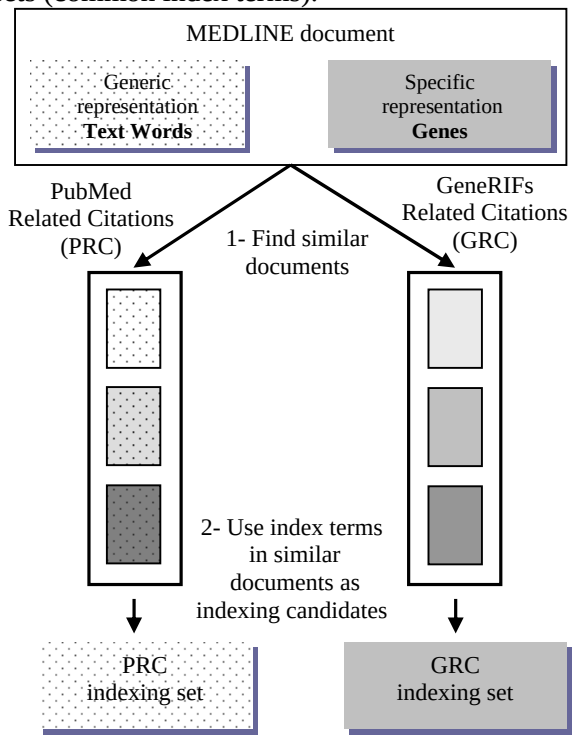


Figure 1: Producing candidate indexing sets with generic and domain-specific representations.

4 Results

Appendix B shows the indexing sets obtained from the GRC and PRC algorithms for a sample citation from the test corpus. Table 2 presents the results of our experiments. For each category of descriptors, the best performance was bolded. It can be observed that in general, the best precision and F1 scores are obtained with the common indexing set, the best recall is obtained with the pooling of indexing sets and the best F3 score is obtained with PRC algorithm, the pooling of indexing sets being a close second.

5 Discussion

5.1 Performance of the methods

As can be seen from the bolded figures in table 2, the best performance is obtained either from the PRC algorithm, or from a combination of PRC and GRC. When indexing methods are combined, it is usually expected that statistical methods will provide the best recall whereas linguistic methods will provide the best precision. Combining complementary methods is then expected to provide the best overall performance. In this context, it seems that the option of pooling the indexing sets should be retained for further experiments. The most significant result of this study is that the pooling of methods achieves a recall of 92% for stand-alone subheading retrieval. While the precision is only 19%, the selection of stand-alone subheadings offered by our methods is nearly exhaustive and it reduces by 70% the size of the list of allowable subheadings that could potentially be used. NLM indexers have declared this could prove very useful to enhance their indexing practice.

In order to qualify the added value of the specific description, we looked at the descriptors that were correctly recommended by GRC and not recommended by PRC. Check Tags (descriptors used to denote the species, age and gender of the subjects discussed in an article) seemed prominent, but only *Human* was significantly recommended correctly more often than it was recommended incorrectly (~2.2 times more correct than incorrect recommendations – 2,712 correct *vs.* 1,250 incorrect). No other descriptor could be identified as being consistently recommended either correctly or incorrectly.

For both methods, filtering the indexing sets according to the number of neighbors that lead to include the indexing terms results in an increase of precision and a loss of recall. The best trade-off (measured by F1) is obtained when indexing terms come from at least three neighbors (data not shown).

5.2 A scale of indexing performance

The problem with evaluating indexing is that, although inter-indexer variability is reduced when a controlled vocabulary is used, indexing is an open cognitive task for which there is no unique "right" solution.

Table 2: performance of the indexing methods on the four categories of descriptors

	SH				**MH**				**SH/MH**				**DESC**			
	P	*R*	*F1*	*F3*	*P*	*R*	*F1*	*F3*	*P*	*R*	*F1*	*F3*	*P*	*R*	*F1*	*F3*
GRC	21	72	32	58	8	49	14	32	3	23	6	14	6	38	10	25
PRC	27	88	41	**72**	13	61	22	**45**	8	56	15	**36**	11	59	18	**41**
Pool	19	**92**	32	67	9	**82**	16	44	5	**62**	9	29	7	**74**	13	38
Common	**36**	68	**47**	62	**22**	27	**24**	27	**18**	17	**17**	17	**21**	23	**22**	23

In practice, this means that there is no ideal unique set of descriptors to use for the indexing of a particular document. Therefore, when comparing an indexing set obtained automatically (e.g. here with the PRC or GRC methods) to a "gold standard" indexing set produced by a trained indexer (e.g. here, NLM indexers) the difference observed can be due to erroneous descriptors produced by the automatic methods. But it is also likely that the automatic methods will produce terms that are semantically close to what the human indexer selected or even relevant terms that the human indexer considered or forgot to select. While evaluation methods to assess the semantic similarity between indexing sets are investigated (Névéol et al. 2006), a consistency study by Funk et al. (1983) can shade some light on inter-indexer consistency in MEDLINE and what range of performance may be expected from automatic systems. In this study, Hooper's consistency (the average proportion of terms in agreement between two indexers) for stand-alone subheadings (SH) was 48.7%. It was 33.8% for pairs (MH/SH) and 48.2% for main headings (MH). In light of these figures, although no direct comparison with the results of our experiment is possible, the precision obtained from the common recommendations (especially for stand-alone subheadings, 36%) seems reasonably useful. Further more, when informally presenting the indexers sample recommendations obtained with these methods, they expressed their interest in the high recall as reviewing a larger selection of potentially useful terms might help them track important descriptors they may not have thought of using otherwise.

In comparison with other research, the results are also encouraging: the recall resulting from either PRC or pooling the indexing sets is significantly better than that obtained by Névéol et al. (2007) on a larger set of MEDLINE 2005 citations – 20% at best for main heading/subheading pairs with a dictionary-based method which consisted in extracting main heading and subheading separately from the citations (using MTI and string matching dictionary entries) before forming all the allowable pairs as recommendations.

5.3 Limitations of the experiment

In the specific description, the mapping between gene names and Entrez Gene IDs only takes human genes into account, which potentially limits the scope of the method, since many more organisms and their genes may be discussed in the literature. In some cases, this limitation can lead to confusion with other organisms. For example, the gene EPO "erythropoietin" is listed in Entrez Gene for 11 organisms including *Homo Sapiens*. With our current algorithm, this gene will be assumed to be a human gene. In the case of PMID 15213094 in our test corpus, the organism discussed in the paper was in fact *Mus Musculus* (common mouse). In this particular case, the check tag *Humans*, which was erroneous, could be found in the candidate indexing set. However,

correct indexing terms could still be retrieved due to the fact that both the human and mouse gene share common functions.

Another limitation is the size of the test corpus, which was limited to less than 4,000 documents.

5.4 Mining the biomedical literature for gene-concept links

Other approaches to gene-keyword mapping exploit the links between genes and diseases or proteins as they are described either in the records of databases such as OMIM or more formally expressed as in the GeneRIF. Substantial work has addressed linking DNA microarray data to keywords in controlled vocabulary such as MeSH (Masys et al. 2001) or characterizing gene clusters with text words from the literature (Liu et al. 2004). However, no normalized "semantic fingerprinting" has been yet produced between controlled sets such as Entrez Gene and MeSH terms.

6 Conclusion and future work

In this paper, we applied a statistical method for indexing documents from the genetics literature. We presented two different document representations, one generic and one specific to the genetics domain. The results bear out our expectations that such statistical methods can also be used successfully to produce recommendations involving subheadings. Furthermore, they yield higher recall than other more linguistic-based methods. In terms of recall, the best results are obtained when the indexing sets from both the specific and generic representations are pooled.

In future work, we plan to refine the algorithm based on the specific method by expending its scope to other organisms than *Homo Sapiens* and to take the gene frequency in the title and abstract of documents into account for the representation. Then, we shall conduct further evaluations in order to observe the impact of these changes, and to verify that similar results can be obtained on a larger corpus.

Acknowledgments

This research was supported in part by an appointment of A. Névéol to the Lister Hill Center Fellows Program sponsored by the National Library of Medicine and administered by the Oak Ridge Institute for Science and Education. The authors would like to thank Halil Kilicoglu for his help with obtaining Entrez Gene IDs from the ABgene output. We also thank Susanne Humphrey and Sonya Shooshan for their insightful comments on the preparation and editing of this manuscript.

References

Alan R. Aronson, Olivier Bodenreider, H. Florence Chang, Susanne M. Humphrey, James G. Mork, Stuart J. Nelson, Thomas C. Rindflesch and W. John Wilbur. 2000. The NLM Indexing Initiative. *Proceedings of the Annual American Medical Informatics Association Symposium.* (AMIA 2000): 17-21.

Alan R. Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proceedings of the Annual AMIA Symposium.* (AMIA 2001):17-21.

Alan R. Aronson, James G. Mork, Cliff W. Gay, Susanne M. Humphrey and William J. Rogers. 2004. The NLM Indexing Initiative's Medical Text Indexer. *Proceedings of Medinfo 2004*: 268-72.

Kate Byrne and Ewan Klein. 2003. Image Retrieval using Natural Language and Content-Based techniques. In Arjen P. de Vries, ed. *Proceedings of the 4th Dutch-Belgian Information Retrieval Workshop (DIR 2003)*:57-62.

Scott Deerwester, Susan Dumais, Georges Furnas, Thomas Landauer and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of American Society for Information Science*, 6 (41):391-407.

Mark E. Funk, Carolyn A. Reid and Leon S. McGoogan. 1983. Indexing consistency in MEDLINE. *Bull. Med. Libr. Assoc.* 71(2):176-183.

Susanne M. Humphrey. 1999. Automatic indexing of documents from journal descriptors: a preliminary investigation. *J Am Soc Inf Sci Technol*. 50(8):661-674

Christian Jacquemin and Evelyne Tzoukermann. 1999. NLP for term variant extraction: Synergy of morphology, lexicon, and syntax. In T. Strzalkowski (Ed.), *Natural language information retrieval* (p. 25-74). Boston, MA: Kluwer.

Won Kim, Alan R. Aronson and W. John Wilbur. 2001. Automatic MeSH term assignment and quality assessment. *Proceedings of the Annual AMIA Symposium*: 319-23.

Ying Liu, Martin Brandon, Shamkant Navathe, Ray Dingledine and Brian J. Ciliax. 2004. Text mining functional keywords associated with genes. *Proceedings of MEDINFO 2004*: 292-296

Daniel R. Masys, John B. Welsh, J. Lynn Fink, Michael Gribskov, Igor Klacansky and Jacques Corbeil. 2001. Use of keyword hierarchies to interpret gene expression patterns. In: *Bioinformatics* 17(4):319-326

Adeline Nazarenko and Touria Ait El Mekki 2005. Building back-of-the-book indexes. In: *Terminology* 11(1):199–224

Aurélie Névéol, Kelly Zeng, Olivier Bodenreider. 2006. Besides precision & recall: Exploring alternative approaches to evaluating an automatic indexing tool for MEDLINE. *Proceedings of the Annual AMIA Symposium*: 589-93.

Aurélie Névéol, Sonya E. Shooshan, Susanne M. Humphrey, Thomas C. Rindflesch and Alan R Aronson. 2007. Multiple approaches to fine-grained indexing of the biomedical literature. *Proceedings of the 12th Pacific Symposium on Biocomputing*. 12:292-303

James Rhodes, Stephen Boyer, Jeffrey Kreulen, Ying Chen, Patricia Ordonez. 2007. Mining Patents Using Molecular Similarity Search. *Proceedings of the 12th Pacific Symposium on Biocomputing*. 12:304-315

Thomas C. Rindflesch and Marcelo Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *J Biomed Inform.* **36(6)**, 462-77

Gerald Salton. 1989. *Automatic text processing : The transformation, analysis, and retrieval of information by computer*. Reading, MA : Addison-Wesley.

Karen Sparck-Jones, Steve Walker and Stephen E. Robertson. 2000. A probalistic model of information retrieval: development and comparative experiments (part 1). *Information Processing and Management*, 36(3):779-808.

Lorraine Tanabe and W. John Wilbur. 2002. Tagging gene and protein names in biomedical text. *Bioinformatics*. 2002 Aug;18(8):1124-32.

Appendix A: Title, abstract and reference indexing set for a sample citation

PubMed ID	15645653
Title	Identification of two novel missense mutations (p.R1221C and p.R1357W) in the ABCC6 (MRP6) gene in a Japanese patient with pseudoxanthoma elasticum (PXE).
Abstract	Pseudoxanthoma elasticum (PXE) is a rare, inherited, systemic disease of elastic tissue that in particular affects the skin, eyes, and cardiovascular system. Recently, the ABCC6 (MRP6) gene was found to cause PXE. A defective type of ABCC6 gene (16pl3.1) was determined in two Japanese patients with PXE. In order to determine whether these patients have a defect in ABCC6 gene, we examined each of 31 exons and flanking intron sequences by PCR methods (SSCP screening and direct sequencing). We found two novel missense variants in exon 26 and 29 in a compound heterozygous state in the first patient. One is a missense mutation (c.3661C>T; p.R1221C) in exon 26 and the other is a missense mutation (c.4069C>T; p.R1357W) in exon 29. These mutations have not been detected in our control panel of 200 alleles. To our knowledge, this is the first report of mutation identification in the ABCC6 gene in Japanese PXE patients. The second patient was homozygous for 2542_2543delG in ABCC6 gene and heterozygous for 6 kb deletion of LDL-R gene. This case is the first report of a genetically confirmed case of double mutations both in PXE and FH loci.
MeSH reference indexing set	Adult Aged Female Humans Japan Multidrug Resistance-Associated Proteins/*genetics *Mutation, Missense Pedigree Pseudoxanthoma Elasticum/*genetics

Appendix B: Sample indexing sets obtained from the GRC and PRC algorithms for a sample citation

PubMed ID	15645653
GRC indexing set[*] (top 15 terms)	Humans (10) Multidrug Resistance-Associated Proteins (9) Mutation (8) Male (7) Female (7) Multidrug Resistance-Associated Proteins/genetics (7) Pseudoxanthoma Elasticum (6) Pseudoxanthoma Elasticum/genetics (6) Pedigree (5) Exons (4) DNA Mutational Analysis (4) Mutation/genetics (4) Adult (4) Introns (3) *Aged* (3)
PRC indexing set[*] (top 15 terms)	Multidrug Resistance-Associated Proteins (10) Multidrug Resistance-Associated Proteins /genetics (10) Pseudoxanthoma Elasticum (10) Pseudoxanthoma Elasticum/genetics (10) Mutation (7) DNA Mutational Analysis (6) Pedigree (5) Genotype (4) Polymorphism, Genetic (4) Alleles (4) Mutation/genetics (3) Haplotypes (3) Models, Genetic (3) Gene Deletion (3) Exons (3)

[*] Terms appearing in the reference set are underlined; the number of neighbors – out of the 10 nearest neighbors – labeled with each term is shown between brackets after the term.

Developing Feature Types for Classifying Clinical Notes

Jon Patrick, Yitao Zhang and **Yefeng Wang**
School of Information Technologies
University of Sydney
NSW 2006, Australia
{jonpat, yitao, ywang1}@it.usyd.edu.au

Abstract

This paper proposes a machine learning approach to the task of assigning the international standard on classification of diseases ICD-9-CM codes to clinical records. By treating the task as a text categorisation problem, a classification system was built which explores a variety of features including negation, different strategies of measuring gloss overlaps between the content of clinical records and ICD-9-CM code descriptions together with expansion of the glosses from the ICD-9-CM hierarchy. The best classifier achieved an overall F_1 value of 88.2 on a data set of 978 free text clinical records, and was better than the performance of two out of three human annotators.

1 Introduction

Despite the rapid progress on text categorisation in the newswire domain, assigning meaningful labels to clinical notes has only recently emerged as a topic for computational linguists although health informatics researchers have been working on the problem for over 10 years. This paper describes constructing classifiers for the Computational Medicine Center's 2007 Medical Natural Language Processing Challenge which aims to assign ICD-9-CM codes to free text radiology reports. (Computational Medicine Center, 2007) It addresses the difficulties of medical text categorisation tasks by incorporating medical negations, term variations, and clues from hierarchy of medical ontologies as additional features.

2 The task of assigning ICD-9-CM codes

The corpus used in this study is a collection of radiology reports from the Cincinnati Children's Hospital Medical Center, Department of Radiology. (Computational Medicine Center, 2007) The data set is divided into a training set and a test set. The training set consists of 978 records and the test set consists of 976 records and 45 ICD-9-CM code. The task was considered as a multi-label text categorisation problem. For each code found in the corpus, we created a separate classifier which makes binary "Yes" or "No" decisions for the target code of a clinical record. Maximum Entropy Modeling (MaxEnt) (Berger et al., 1996) and Support Vector Machine (SVM) (Vapnik, 1995) were used to build the classifiers in our solution.

3 Features

A variety of features were developed to represent what we believed were the important determiners of the ICD-9-CM codes.

Bag-of-words (BOW) features: include only unigrams and bigrams in the text.

Negation features: were used in the classification system to capture the terms that are negated or uncertain, for example "pneumonia" vs "no evidence of pneumonia". We created a negation-finding system which uses an algorithm similar to (Chapman et al., 2001) to identify the negation phrase and the scope of negations.

Gloss matching feature: The ICD-9-CM provides detailed text definition for each code. This section explores different strategies for measuring gloss

BioNLP 2007: Biological, translational, and clinical language processing, pages 191–192,
Prague, June 2007.

Name	Description	P	R	F_1
S0	BOW baseline	83.9	78.4	81.1
S1	S0 + negation	88.5	78.2	83.0
S2	S1 + gloss matching	89.2	80.6	84.7
S3	feature engineering	89.7	86.0	87.8
S4	S3 + low-freq	89.7	86.9	88.2

Table 1: Experiment results for all ICD-9-CM codes

matchings between the content of a clinical record and the definition of an ICD-9-CM code.

Feature engineering: In experiments with a uniform set of feature types for all ICD-9-CM codes, we noticed that different codes tend to have a preference for different combinations of feature types. Therefore, different combinations of feature types for each individual code were used. The intuition is to explore different combination of feature types quickly instead of doing further feature selection procedures. The system trained on the best combination of feature types are reported as the final results for the target code.

Low frequency codes modeling: A rule-based system was also used to model low frequency ICD-9-CM codes which have only one occurrence in the corpus, or have achieved F_1 value of 0.0 by machine learning. The system assigns a low frequent code to a clinical record if the content of the record matches the words of the code definition.

4 Result

Table 1 shows the experiment results. Since the gold-standard annotation of the test dataset has not been released so far, the experiment was done on the 978 documents training dataset using 10-fold cross-validation. [1] The baseline system S0 was created using only BOW features. Adding negation features gives S1 an improvement of 1.9% on F_1 score. The gloss matching features gives a further increase of 1.7% on F_1 score.

In order to understand more about the ICD-9-CM code assignment task, this section evaluates the performance of the three annotators. Table 2 compares the performance of each annotator to the gold-standard codes. The item "N" in Table 2 stands for the total number of ICD-9-CM codes which an annotator has assigned to the whole corpus.

Name	P	R	F_1	N
company1	78.3	89.8	83.7	1397
company2	82.6	95.2	88.5	1404
company3	90.4	75.0	82.0	1011
S4	89.7	86.9	88.2	1180

Table 2: Performances of Annotators

5 Conclusion

This paper presents an approach to the problem of assigning ICD-9-CM codes to free text medical records. We created a classification system which consists of multiple machine-learned classifiers on high-frequency codes, and a rule-based modeling module of low-frequency codes. By incorporating negations and a variety of gloss matching features, we successfully outperformed the baseline with only bag-of-words features by 7.1% on F_1 value. The best reported score is also considered as comparable to the performance of the best human annotator. We also consider the way our system selected the best combination of feature types for each individual ICD-9-CM code has a major contribution to the classification task of clinical records.

References

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

W.W. Chapman, W. Bridewell, P. Hanbury, G.F. Cooper, and B.G. Buchanan. 2001. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34(5):301–310.

Computational Medicine Center. 2007. 2007 Medical Natural Language Processing Challenge. http://www.computationalmedicine.org/challenge/.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

[1] The official score of our system on the test dataset is $F_1 = 86.76$ which was ranked 7th among 44 systems. See http://www.computationalmedicine.org/challenge/res.php

Quantitative Data on Referring Expressions in Biomedical Abstracts

Michael Poprat **Udo Hahn**
Jena University Language and Information Engineering (JULIE) Lab
Fürstengraben 30, 07743 Jena, Germany
{poprat|hahn}@coling-uni-jena.de

Abstract

We report on an empirical study that deals with the quantity of different kinds of referring expressions in biomedical abstracts.

1 Problem Statement

One of the major challenges in NLP is the resolution of referring expressions. Those references can be established by repeating tokens or by pronominal, nominal and bridging anaphora. Experimental results show that pronominal anaphora are easier to resolve than nominal ones because the resolution of nominal anaphora requires an IS-A-taxonomy as knowledge source. The resolution of bridging anaphora, however, proves to be awkward because encyclopedic knowledge is necessary.[1] But in practice, are all of these phenomena equally important? A look at the publications reveals that a comprehensive overview of the quantity and distribution of referring expressions in biomedical abstracts is still missing. Nevertheless, some scattered data can be found: Castaño et al. (2002) state that 60 of 100 anaphora are nominal anaphora. Sanchez et al. (2006) confirm this proportion (24 pronominal and 50 nominal anaphora in 74 anaphoric expressions). Kim and Park (2004), however, detect 53 pronominal and 26 nominal anaphora in 87 anaphoric expressions. But Gawronska and Erlendsson (2005), on the other hand, claim that pronominal anaphora are rare and nominal anaphora are predominant. Studies on bridging anaphora in the biomedical domain are really still missing. Only Cimiano (2003) states that 10% of definite descriptions are bridging anaphora.

This contradictoriness and the lack of statistics on referring expressions induced us to collect our own data in order to obtain a consistent and meaningful overview. This picture helps to decide where to start if one wants to build a resolution component for the biomedical domain.

2 Empirical Study

For our study we selected articles from MEDLINE for stem cell transplantation and gene regulation. Out of these articles, 11 stem cell abstracts and 9 gene regulation abstracts ($\sim$ 12,000 tokens) were annotated by a team of one biologist and one computational linguist. The boundaries for annotations were neither limited to nominal phrases (NPs) nor on their heads because NPs in biomedical abstracts are often complex and hide relations between nouns (e.g., a *"p53 protein"* is a protein called *"p53"*, a *"p53 gene"* is a gene that codes the *"p53 protein"* and a *"p53 mutation"* is a mutation in the *"p53 gene"*). Furthermore, we annotated anaphoric expressions referring to biomedical entities and to processes.

We distinguished the following referring expressions: As repetitions, we counted string-identical, string-variants and abbreviated token sequences in NPs, identical in their meaning (e.g. *"Mesenchymal stem cells"* - *"MSCs"* - *"MSC inhibitory effect"*). For the time being, modifiers have not been considered. Anaphora comprise pronominal[2], nominal (IS-A relations, e.g., *"B-PLL"* IS-

[1] However, even the resolution of pronouns can benefit from extra-textual information (Castaño et al., 2002).

[2] Without "we" as it always refers to the authors.

BioNLP 2007: Biological, translational, and clinical language processing, pages 193–194,
Prague, June 2007.

Type of Referring Expression	Number
Repetitions	388
Pronominal Anaphora	48 (sent. internal) 6 (sent. external)
Nominal Anaphora	79
Bridging Anaphora	42
Subgrouping Anaphora	91
all	**654**

Table 1: Number of Referring Expressions

A *"B-cell malignancy"*) and bridging anaphora (all other semantic relations, e.g., *"G(1) progression"* PART-OF-PROCESS *"M-G(1) transition"*). Furthermore, we detected a high number of subgrouping anaphora that often occur when a group of entities (e.g., *"Vascular endothelial growth factor receptors"*) are mentioned first and certain subgroups (e.g., *"VEGFR1"* etc.) are discussed later.

In our abstracts we detected 654 referring expressions (see Table 1). Repetitions are predominant with 59%. Within the group of 266 anaphora, subgrouping anaphora contributed with 34%, nominal anaphora with 30%, pronominal anaphora with 20% and bridging anaphora with only 16%. The most common bridging relations were PART-OF-AMOUNT (14) and PART-OF (11). The remaining 17 are held by 8 other semantic relations such as RESULTS-FROM, MUTATED-FROM, etc.

3 Open Issues and Conclusion

In biomedical abstracts we are confronted with numerous repetitions, mainly containing biomedical entities. Their reference resolution within an abstract seems to be easy at first glance by just comparing strings and detecting acronyms. Some examples will show that this is tricky, though: In *"The VEGFR3-transfected ECs exhibited high expression level of LYVE-1."*, this statement on ECs only holds if the modifier *"VEGFR3-transfected"* is taken into account. Furthermore, transfected ECs are not identical with non-transfected ECs which would be the result if considering NP heads only. But not every modifier influences an identity relation. For example, the purification in *"...when priming with purified CD34(+) cells"* has no influence on the CD34(+) cells and statements about these cells keep their generality. A classification of such modifiers adding information with or without influencing the semantics of the modified expression must be made. Hence, we have to be careful with assumed repetitions and we have to handle all kinds of modifiers.

In this study we present the first comprehensive overview of various kinds of referring expressions that occur in biomedical abstracts. Although our corpus is still small, we could observe the strong tendency that repetitions play a major role (20 per abstract). Anaphora occur less frequently (13 per abstract). For a sound semantic interpretation, both types must be handled. For knowledge-intensive anaphora resolution, the existing biomedical resources must be reviewed for adequacy. To the best of our knowledge, although dominant in our study, subgrouping anaphora have not been considered in any anaphora resolution systems and suitable resolution strategies must be found. The annotation process (with more than one annotation team) will be continued. The main result of this study, however, is the observation that modifiers play an important role for referencing. Their treatment for semantic interpretation requires further investigations.

Acknowledgements: We thank Belinda Würfel for her annotation work. This study was funded by the EC (BOOTStrep, FP6-028099), and by the German Ministry of Education and Research (StemNet, 01DS001A - 1C).

References

J. Castaño, J. Zhang, and J. Pustejovsky. 2002. Anaphora resolution in biomedical literature. In *Proc. of the Symp. on Reference Resolution for NLP*.

P. Cimiano. 2003. On the research of bridging references within information extraction systems. Diploma thesis, University of Karlsruhe.

B. Gawronska and B. Erlendsson. 2005. Syntactic, semantic and referential patterns in biomedical texts: Towards in-depth comprehension for the purpose of bioinformatics. In *Proc. of the 2nd Workshop on Natural language Understanding and Cognitive science*, pages 68–77.

J.-J. Kim and J. C. Park. 2004. BioAR: Anaphora resolution for relating protein names to proteome database entries. In *Proc. of the ACL 2004: Workshop on Reference Resolution and its Applications*, pages 79–86.

O. Sanchez, M. Poesio, M. A. Kabadjov, and R. Tesar. 2006. What kind of problems do protein interactions raise for anaphora resolution? A preliminary analysis. In *Proc. of the 2nd SMBM 2006*, pages 109–112.

Discovering contradicting protein-protein interactions in text

Olivia Sanchez-Graillet
Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, U.K.
osanch@essex.ac.uk

Massimo Poesio
Univ. of Essex, Wivenhoe Park, Colchester CO4 3SQ, U.K.
DIT and Center for Mind/Brain Sciences, Univ. of Trento, Via Sommarive 14 I-38050 POVO (TN) - Italy
poesio@essex.ac.uk

1 Introduction

In biomedical texts, contradictions about protein-protein interactions (PPIs) occur when an author reports observing a given PPI whereas another author argues that very same interaction does not take place: e.g., when author X argues that "protein A interacts with protein B" whereas author Y claims that "protein A does not interact with B". Of course, merely discovering a potential contradiction does not mean the argument is closed as other factors may have caused the proteins to behave in different ways. We present preliminary work towards the automatic detection of potential contradictions between PPIs from text and an agreement experimental evaluation of our method.

2 Method

Our method consists of the following steps: i) extract positive and negative cases of PPIs and map them to a semantic structure; ii) compare the pairs of PPIs structures that contain similar canonical protein names iii) apply an inference method to the selected pair of PPIs.

We extract positive and negative cases of PPIs by applying our system (Sanchez & Poesio, submitted). Our system considers proteins only as well as events where only one protein participates (e.g. "*PI-3K activity"*). The system produces the semantic interpretation shown in Table 1. We manually corrected some of the information extracted in order to compare exclusively our inference method with human annotators.

The decision to determine if a C-PPI holds is given by the context. This context is formed by the combination of semantic components such as PPI polarity, verb direction, and manner polarity.

P1	Canonical name of the first participant protein
P2	Canonical name of the second participant protein.
Cue-word	Word (verbs or their nominalizations) expressing a PPI (e.g. interact, interaction, activate, activation, etc.).
Semantic Relation	Categories in which cue-words are grouped according to their similar effect in an interaction. (See Table 2).
Polarity	Whether the PPI is positive or negative
Direction	Direction of a relation according to the effect that a protein causes on other molecules in the interaction. (See Table 3)
Manner	Modality expressed by adverbs or adjectives (e.g. directly, weakly, strong, etc.)
Manner Polarity	Polarity assigned to manner according to the influence they have on the cue-word (see Table 4)

Table 1. Semantic structure of a PPI

Semantic Relation	*Verbs/nouns examples*
Activate	Activat (e, ed,es,or,ion), transactivat (e,ed,es,ion)
Inactivate	decreas (e,ed,es), down-regulat(e,ed,es,ion)

Table 2. Example of semantic verb relations

+	-	*Neutral*
Activate, Attach	Inactivate	Substitute, React
Create bond	Break bond	Modify, Cause
Generate	Release	Signal, Associate

Table 3. Directions of semantic relations

Polarity	*Word*
(+) 1	strong(ly), direct(ly), potential(y), rapid(ly)
(-) 0	hardly, indirect(ly), negative(e,ly)

Table 4. Example of manner polarity

Manner polarity is neutral (2) if the manner word is not included in the manner polarity table or if no manner word affects the cue-word.

The method first obtains what we call "PPI state" of each PPI. The PPI state is obtained in two steps that follow decision tables[1]: a) the values for

[1] Some decision tables are omitted due to space reasons.

BioNLP 2007: Biological, translational, and clinical language processing, pages 195–196,
Prague, June 2007.

the combination of the verb direction and the manner polarity (DM) of each PPI; b) then, the DM value and the polarity of the corresponding PPI are evaluated.

Second, the method compares the PPI states of both PPIs as shown in Table 5.

State1	*Sstate2*	*Result*	*State1*	*State2*	*Result*
0	0	NC	3	3	U
0	1	C	0	4	C
0	3	U	1	4	C
1	1	NC	3	4	C
1	3	U			

Table 5. Decision table for results[2]

The following example illustrates our method. The table below shows two sentences taken from different documents.

Document 1	*Document 2*
Cells treated with hyperosmolar stress, UV-C, IR, or a cell-permeable form of ceramide, *C2 ceramide*, rapidly down-regulated *PI(3)K* activity to 10%-30% of the activity found in serum-stimulated control cells…	And fourth, *C2-ceramide* did not affect the amount of *PI 3-kinase* activity in anti-IRS-1 precipitates.

The semantic structures corresponding to these sentences are shown in the next table.

	DocA	*DocB*
P1	C2-ceramide	C2-ceramide
P2	PI-3K	PI-3K
Cue	down-regulate	affect
Semantic relation	Inactivate	Cause
Polarity	positive	negative
Direction	negative	neutral
Manner	rapidly	--
Manner polarity	positive	neutral

The decision tables produced for this example are the following[3].

PPI	*Direction*	*Manner*	*DM*
A	- (0)	+ (1)	- (0)
B	N (2)	N (2)	U (3)

PPI	*Polarity*	*DM*	*State*
A	+ (1)	- (0)	- (0)
B	- (0)	U(3)	NN (4)

PPIA state	*PPIB state*	*Result*
-(0)	NN (4)	**Contradiction**

The result obtained is "Contradiction".

3 Agreement experiment

As a way of evaluation, we compared agreement between our method and human annotators by using the kappa measure (Siegel and Castellan, 1998). We elaborated a test containing only of 31 pairs of sentences (*JBC* articles) since this task can be tiring for human annotators.

The test consisted on classifying the pairs of sentences into three categories: contradiction (C), no contradiction (NC) and unsure (U). The values of kappa obtained are presented in the following table.

Groups	*Kappa*
Biologists only	0.37
Biologists and our method	0.37
Non-biologists only	0.22
Non-biologists and our method	0.19

Table 6 Agreement values

Biologists mainly justified their answers based on biological knowledge (e.g. methodology, organisms, etc.) while non-biologists based their answers on syntax.

4 Conclusions

We have presented a simple method to detect potential contradictions of PPIs by using context expressed by semantics and linguistics constituents (e.g. modals, verbs, adverbs, etc). Our method showed to perform similarly to biologists and better than non-biologists. Interestingly, biologists concluded that C-PPIs are rarely found; nevertheless, the cases found may be highly significant.

Continuing with our work, we will try our system in a larger set of data.

References

Sanchez,O and Poesio,M. (Submitted). Negation of protein-protein interactions: analysis and extraction.

Siegel, S. and Castellan, N.J. (1998). Nonparametric statistics for the behavioral sciences. 2nd. edition, McGraw-Hill.

[2] Result values: contradiction (C), no contradiction (NC) and unsure (U).

[3] The values included in the tables are: positive=1, negative=0, neutral=2, unsure=3, and negative-neutral=4.

Marking Time in Developmental Biology

Gail Sinclair and Bonnie Webber
School of Informatics
University of Edinburgh
Edinburgh EH8 9LW
c.g.sinclair@ed.ac.uk, bonnie@inf.ed.ac.uk

1 Introduction

In developmental biology, to support reasoning about cause and effect, it is critical to link genetic pathways with processes at the cellular and tissue level that take place beforehand, simultaneously or subsequently. While researchers have worked on resolving with respect to absolute time, events mentioned in *medical* texts such as clinical narratives (e.g. Zhou et al, 2006), events in developmental biology are primarily resolved relative to other events.

In this regard, I am developing a system to extract and time-stamp event sentences in articles on developmental biology, looking beyond the sentence that describes the event and considering ranges of times rather than just single timestamps.

I started by creating four gold standard corpora for documents, event sentences, entities and timestamped events (for future public release). These datasets are being used to develop an automated pipeline to (1) retrieve relevant documents; (2) identify sentences within the documents that describe developmental events; and (3) associate these events with the developmental stage(s) that the article links them with or they are known to be linked with through prior knowledge.

Different types of evidence are used in each step. For determining the relevant developmental stage(s), the text surrounding an event-containing sentence is an efficient source of temporal grounding due of its immediate accessibility. However, this does not always yield the correct stage and other sources need to be used. Information within the sentence, such as the entities under discussion, can also be used to help with temporal grounding using mined background knowledge about the period of existence of an entity.

2 Creation of Datasets

In creating the four new data sets mentioned above, I annotated 1200 documents according to relevance to murine kidney development. From 5 relevant documents, 1200 sentences were annotated as to whether they contained an event description. (Two annotators - one biologist, one computer scientist - achieved an inter-annotator agreement kappa score of 95%.) A sentence is considered a positive one if it contains a description of the following event types:

- molecular expression within tissue/during process/at stage X (molecular event)
- tissue process, i.e. what forms from what (tissue event)
- requirement of a molecule for a process (molecular or tissue event)
- abnormality in a process/tissue/stage (molecular or tissue event)
- negation of the above e.g. was not expressed, did not form, formed normally (molecular or tissue event).

A negative sentence is one that does not fall under at least one of the above categories.

In addition, 6 entities (*tissue, process, species, stage, molecule and event verb*) were annotated in 1800 sentences (1200 described above + 600 from

BioNLP 2007: Biological, translational, and clinical language processing, pages 197–198,
Prague, June 2007.

relevant documents not yet annotated at sentence level) and 347 entity-annotated positive event sentences were marked with their associated developmental stage.

Example: *At E11, the integrin α8 subunit was expressed throughout the mesenchyme of the nephrogenic cord.* Entities annotated: *E11*(stage), *integrin α8* (molecule), *expressed* (event verb), *mesenchyme of the nephrogenic cord* (tissue).

3 Evidence for Temporal Resolution

Developmental biology is not as concerned with the absolute time of events in a specific embryo as it is with events that generally happen under the same circumstances in developmental time. These are referred to with respect to *stages* from conception to birth. The evidence sufficient to resolve the developmental stage of an event sentence can come from many places. The two significant areas of evidence are *local context* (i.e. surrounding text) and *prior* (i.e. background) knowledge.

Local context can further be classified as:

- **explicit**: evidence of stage is mentioned within current (event) sentence,
- **previous sentence**: evidence is found in sentence immediately previous to current sentence,
- **following sentence**: evidence is found in sentence immediately following current sentence,
- **current paragraph**: evidence is found in paragraph containing current sentence but not in adjacent sentences,
- **referenced to figure**: evidence is found in figure legend referenced in current sentence.

Evidence Source	# Event Sentences
Explicitly Stated	48
Immed Prev Sentence	7
Following Sentence	1
Current Paragraph	19
Referenced Figure Legend	38
Within Figure Legend	43
Time Irrelevant	65
Prior Knowledge	126
Total	347

When local context does not provide evidence, **prior knowledge** can be used about when entities mentioned within the sentence normally appear within development. Event sentences can also be **irrelevant** of individual time ranges and apply to the whole of development. The table above shows the frequency with which each evidence type is used to resolve developmental stage.

4 Experiments

Event sentence retrieval experiments (using separate training and test data) resulted in a F-score of 72.3% and 86.6% for Naive Bayes and rule-based classification approaches respectively (relying upon perfect entity recognition). A baseline method (classifying all sentences as positive) achieves 58.4% F-score.

Experiments were also carried out to assign developmental stage to sentences already known to contain events. The baseline approach is to use the last mentioned stage in the text and any methods developed should score higher than this baseline. Rules were developed to assign developmental stage based on the knowledge gained from two fifths of the investigations into temporal evidence described above. The other three fifths were annotated after the rules had been defined. Precision scores for all 347 sentences can be seen in the following table with the *Naive* method representing the baseline and *Local* representing the use of rules.

Paper	Naive Prec.	Local Prec.
1	75.7	97.3
2	89.6	90.9
3	89.1	100
4	95.6	92.3
5	95.5	91.3
Average	89.1	94.5

Experiments are currently ongoing into exploiting the use of background knowledge of the developmental processes and tissues mentioned within event descriptions in order to assign developmental stage to events sentences not already assigned by the local context rules and to increase confidence in those stages already assigned.

References

L. Zhou, G. B. Melton, S. Parsons and G Hripcsak, A temporal constraint structure for extracting temporal information from clinical narrative, J Biomed Inf 39(4), Aug 2006, 424-439

Evaluating and combining biomedical named entity recognition systems

Andreas Vlachos
William Gates Building
Computer Laboratory
University of Cambridge
av308@cl.cam.ac.uk

Abstract

This paper is concerned with the evaluation of biomedical named entity recognition systems. We compare two such systems, one based on a Hidden Markov Model and one based on Conditional Random Fields and syntactic parsing. In our experiments we used automatically generated data as well as manually annotated material, including a new dataset which consists of biomedical full papers. Through our evaluation, we assess the strengths and weaknesses of the systems tested, as well as the datasets themselves in terms of the challenges they present to the systems.

1 Introduction

The domain of biomedical text mining has become of importance for the natural language processing (NLP) community. While there is a lot of textual information available in the domain, either in the form of publications or in model organism databases, there is paucity in material annotated explicitly for the purpose of developing NLP systems. Most of the existing systems have been developed using data from the newswire domain. Therefore, the biomedical domain is an appropriate platform to evaluate existing systems in terms of their portability and adaptability. Also, it motivates the development of new systems, as well as methods for developing systems with these aspects in focus in addition to the performance.

The biomedical named entity recognition (NER) task in particular has attracted a lot of attention from the community recently. There have been three shared tasks (BioNLP/NLPBA 2004 (Kim et al., 2004), BioCreative (Blaschke et al., 2004) and BioCreative2 (Krallinger and Hirschman, 2007)) which involved some flavour of NER using manually annotated training material and fully supervised machine learning methods. In parallel, there have been successful efforts in bootstrapping NER systems using automatically generated training material using domain resources (Morgan et al., 2004; Vlachos et al., 2006). These approaches have a significant appeal, since they don't require manual annotation of training material which is an expensive and lengthy process.

Named entity recognition is an important task because it is a prerequisite to other more complex ones. Examples include anaphora resolution (Gasperin, 2006) and gene normalization (Hirschman et al., 2005). An important point is that until now NER systems have been evaluated on abstracts, or on sentences selected from abstracts. However, NER systems will be applied to full papers, either on their own or in order to support more complex tasks. Full papers though are expected to present additional challenges to the systems than the abstracts, so it is important to evaluate on the former as well in order to obtain a clearer picture of the systems and the task (Ananiadou and McNaught, 2006).

In this paper, we compare two NER systems in a variety of settings. Most notably, we use automatically generated training data and we evaluate on abstracts as well as a new dataset consisting of full papers. To our knowledge, this is the first evaluation of biomedical NER on full paper text instead of

BioNLP 2007: Biological, translational, and clinical language processing, pages 199–200,
Prague, June 2007.

abstracts. We assess the performance and the portability of the systems and using this evaluation we combine them in order to take advantage of their strengths.

2 Named entity recognition systems

This section presents the two biomedical named entity recognition systems used in the experiments of Section 4. Both systems have been used successfully for this task and are domain-independent, i.e. they don't use features or resources that are tailored to the biomedical domain.

2.1 Hidden Markov Model

The first system used in our experiments was the HMM-based (Rabiner, 1990) named entity recognition module of the open-source NLP toolkit LingPipe[1]. It is a hybrid first/second order HMM model using Witten-Bell smoothing (Witten and Bell, 1991). It estimates the following joint probability of the current token x_t and label y_t conditioned on the previous label y_{t-1} and previous two tokens x_{t-1} and x_{t-2}:

$$P(x_t, y_t | y_{t-1}, x_{t-1}, x_{t-2}) \quad (1)$$

Tokens unseen in the training data are passed to a morphological rule-based classifier which assigns them to predefined classes according to their capitalization and whether they contain digits or punctuation. In order to use these classes along with the ordinary tokens, during training a second pass over the training data is performed in which tokens that appear fewer times than a given threshold are replaced by their respective classes. In our experiments, this threshold was set experimentally to 8. Vlachos et al. (2006) employed this system and achieved good results on bootstrapping biomedical named entity recognition. They also note though that due to its reliance on seen tokens and the restricted way in which unseen tokens are handled its performance is not as good on unseen data.

2.2 Conditional Random Fields with Syntactic Parsing

The second NER system we used in our experiments was the system of Vlachos (2007) that participated in the BioCreative2 Gene Mention task (Krallinger and Hirschman, 2007). Its main components are the Conditional Random Fields toolkit MALLET[2] (McCallum, 2002) and the RASP syntactic parsing toolkit[3] (Briscoe et al., 2006), which are both publicly available.

Conditional Random Fields (CRFs) (Lafferty et al., 2001) are undirected graphical models trained to maximize the conditional probability of the output sequence given the inputs, or, in the case of token-based natural language processing tasks, the conditional probability of the sequence of labels y given a sequence of tokens x. Like HMMs, the number of previous labels taken into account defines the order of the CRF model. More formally:

$$P(y|x) = \frac{1}{Z(x)} exp\{\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(y, x_t)\} \quad (2)$$

In the equation above, $Z(x)$ is a normalization factor computed over all possible label sequences, f_k is a feature function and λ_k its respective weight. y represents the labels taken into account as context and it is defined by the order of the CRF. For a n-th order model, y becomes $y_t, y_{t-1}..., y_{t-n}$. It is also worth noting that x_t is the feature representation of the token in position t, which can include features extracted by taking the whole input sequence into account, not just the token in question. The main advantage is that as a conditionally-trained model CRFs do not need to take into account dependencies in input, which as a consequence, allows the use of features dependent on each other. Compared to HMMs, their main disadvantage is that during training, the computation time required is significantly longer. The interested reader is referred to the detailed tutorial of Sutton & McCallum (2006).

Vlachos (2007) used a second order CRF model combined with a variety of features. These can be divided into simple orthographic features and in

[1] http://www.alias-i.com/lingpipe. The version used in the experiments was 2.1.

[2] http://mallet.cs.umass.edu/index.php/Main_Page

[3] http://www.informatics.susx.ac.uk/research/nlp/rasp/

those extracted from the output of the syntactic parsing toolkit. The former are extracted for every token and they are rather common in the NER literature. They include the token itself, whether it contains digits, letters or punctuation, information about capitalization, prefixes and suffixes.

The second type of features are extracted from the output of RASP for each sentence. The part-of-speech (POS) tagger was parameterized to generate multiple POS tags for each token in order to ameliorate unseen token errors. The syntactic parser uses these sequences of POS tags to generate parses for each sentence. The output is in the form of grammatical relations (GRs), which specify the links between the tokens in the sentence accoring to the syntactic parser and they are encoded using the SciXML format (Copestake et al., 2006). From this output, for each token the following features are extracted (if possible):

- the lemma and the POS tag(s) associated with the token
- the lemmas for the previous two and the following two tokens
- the lemmas of the verbs to which this token is subject
- the lemmas of the verbs to which this token is object
- the lemmas of the nouns to which this token acts as modifier
- the lemmas of the modifiers of this token

Adding the features from the output of the syntactic parser allows the incorporation of features from a wider context than the two tokens before and after captured by the lemmas, since GRs can link tokens within a sentence independently of their proximity. Also, they result in more specific features, since the relation between two tokens is determined. The CRF models in the experiments of Section 4 were trained until convergence.

It must be mentioned that syntactic parsing is a complicated task and therefore feature extraction on its output is likely to introduce some noise. The RASP syntactic parser is domain independent but it has been developed using data from general English corpora mainly, so it is likely not to perform as well in the biomedical domain. Nevertheless, the results of the system in the BioCreative2 Gene Mention task suggest that the use of syntactic parsing features improve performance. Also, despite the lack of domain-specific features, the system is competitive with other systems, having performance in the second quartile of the task. Finally, the BIOEW scheme (Siefkes, 2006) was used to tag the tokenized corpora, under which the first token of a multitoken mention is tagged as B, the last token as E, the inner ones as I, single token mentions as W and tokens outside an entity as O.

3 Corpora

In our experiments we used two corpora consisting of abstracts and one consisting of full papers. One of the abstracts corpora was automatically generated while the other two were manually annotated. All three were created using resources from FlyBase[4] and they are publicly available[5].

The automatically generated corpus was created in order to bootstrap a gene name recognizer in Vlachos & Gasperin (2006). The approach used was introduced by Morgan et al (2004). In brief, the abstracts of 16,609 articles curated by FlyBase were retrieved and tokenized by RASP (Briscoe et al., 2006). For each article, the gene names and their synonyms that were recorded by the curators were annotated automatically in its abstract using longest-extent pattern matching. The pattern matching is flexible in order to accommodate capitalization and punctuation variations. This process resulted in a large but noisy dataset, consisting of 2,923,199 tokens and containing 117,279 gene names, 16,944 of which are unique. The noise is due to two reasons mainly. First, the lists constructed by the curators for each paper are incomplete in two ways. They don't necessarily contain all the genes mentioned in an abstract because not all genes are always curated and also not all synonyms are recorded, thus resulting in false negatives. The other cause is the overlap between gene names and common English words or biomedical terms, which results in false positives for

[4] http://www.flybase.net/

[5] http://www.cl.cam.ac.uk/ nk304/Project_Index/#resources

abstracts with such gene names.

The manually annotated corpus of abstracts was described in Vlachos & Gasperin (2006). It consists of 82 FlyBase abstracts that were annotated by a computational linguist and a FlyBase curator. The full paper corpus was described in Gasperin et al. (2007). It consists of 5 publicly available full papers which were annotated by a computational linguist and a FlyBase curator with named entities as well as anaphoric relations in XML. To use it for the gene name recognition experiments presented in this paper, we converted it from XML to IOB format keeping only the annotated gene names.

	noisy abstracts	golden abstracts	full papers
abstracts / papers	16,609	82	5
sentences	111,820	600	1,220
tokens	2,923,199	15,703	34,383
gene names	117,279	629	2,057
unique gene names	16,944	326	336
unique non-gene tokens	60,943	3,018	4,113

Table 1: Statistics of the datasets

The gene names in both manually created corpora were annotated using the guidelines presented in Vlachos & Gasperin (2006). The main idea of these guidelines is that gene names are annotated anywhere they are encountered in the text, even when they are used to refer to biomedical entities other than the gene itself. The distinction between the possible types of entities the gene name can refer to is performed at the level of the shortest noun phrase surrounding the gene name. This resulted in improved inter-annotator agreement (Vlachos et al., 2006).

Statistics on all three corpora are presented in Table 1. From the comparisons in this table, an interesting observation is that the gene names in full papers tend to be repeated more frequently than the gene names in the manually annotated abstracts (6.1 compared to 1.9 times respectively). Also, the latter contain approximately 2 unique gene names every 100 tokens while the full papers contain just 1. This evidence suggests that annotating abstracts is more likely to provide us with a greater variety of gene names. Interestingly, the automatically annotated abstracts contain only 0.6 unique gene names every 100 tokens which hints at inclusion of false negatives during the annotation.

Another observation is that, while the manually annotated abstracts and full papers contain roughly the same number of unique genes, the full papers contain 36% more unique tokens that are not part of a gene name ("unique non-gene tokens" in Table 1). This suggests that the full papers contain a greater variety of contexts, as well as negative examples, therefore presenting greater difficultiy to a gene name recognizer.

4 Experiments

We ran experiments using the two NER systems and the three datasets described in Sections 2 and 3. In order to evaluate the performance of the systems, apart from the standard recall, precision and F-score metrics, we measured the performance on seen and unseen gene names independently, as suggested by Vlachos & Gasperin (2006). In brief, the gene names that are in the test set and the output generated by the system are separated according to whether they have been encountered in the training data as gene names. Then, the standard recall, precision and F-score metrics are calculated for each of these lists independently.

		HMM	CRF+RASP
overall	Recall	75.68	63.43
	Precision	89.14	90.89
	F-score	81.86	74.72
seen genes	Recall	94.48	76.32
	Precision	93.62	95.4
	F-score	94.05	84.80
unseen genes	Recall	33.51	34.54
	Precision	68.42	73.63
	F-score	44.98	47.02
seen genes		435	
unseen genes		194	

Table 2: Results on training on noisy abstracts and testing on manually annotated abstracts

		HMM	CRF+RASP
overall	Recall	58.63	61.40
	Precision	80.56	89.19
	F-score	67.87	72.73
seen genes	Recall	89.82	72.51
	Precision	87.83	94.82
	F-score	88.81	82.18
unseen genes	Recall	35.12	53.03
	Precision	69.48	84.05
	F-score	46.66	65.03
seen genes		884	
unseen genes		1173	

Table 3: Results on training on noisy abstracts and testing on full papers

Tables 2 and 3 report in detail the performance of the two systems when trained on the noisy abstracts and evaluated on the manually annotated abstracts and full papers respectively. As it can be seen, the performance of the HMM-based NER system is better than that of CRF+RASP when evaluating on abstracts and worse when evaluating on full papers (81.86 vs 74.72 and 67.87 vs 72.73 respectively).

Further analysis of the performance of the two systems on seen and unseen genes reveals that this result is more likely to be due to the differences between the two evaluation datasets and in particular the balance between seen and unseen genes with respect to the training data used. In both evaluations, the performance of the HMM-based NER system is superior on seen genes while the CRF+RASP system performs better on unseen genes. On the abstracts corpus the performance on seen genes becomes more important since there are more seen than unseen genes in the evaluation, while the opposite is the case for the full paper corpus.

The difference in the performance of the two systems is justified. The CRF+RASP system uses a complex but more general representation of the context based on the features extracted from the output of syntactic parser, namely the lemmas, the part-of-speech tags and the grammatical relationships, while the HMM-based system uses a simple morphological rule-based classifier. Also, the CRF+RASP system takes the two previous labels into account, while the HMM-based only the previous one. Therefore, it is expected that the former has superior performance on unseen genes. This difference between the CRF+RASP and the HMM-based system is substantially larger when evaluating on full papers (65.03 versus 46.66 respectively) than on abstracts (47.02 versus 44.98 respectively). This can be attributed to the fact that the training data used is generated from abstracts and when evaluating on full papers the domain shift can be handled more efficiently by the CRF+RASP system due to its more complex feature set.

However, the increased complexity of the CRF+RASP system renders it more vulnerable to noise. This is particularly important in these experiments because we are aware that our training dataset contains noise since it was automatically generated. This noise is in addition to that from inaccurate syntactic parsing employed, as explained in Section 2.2. On the other hand, the simpler HMM-based system is likely to perform better on seen genes, whose recognition doesn't require complex features.

We also ran experiments using the manually annotated corpus of abstracts as training data and evaluated on the full papers. The results in Table 4 confirmed the previous assessment, that the performance of the CRF+RASP system is better on the unseen genes and that the HMM-based one is better on seen genes. In this particular evaluation, the small number of unique genes in the manually annotated corpus of abstracts results in the majority of gene names being unseen in the training data, which favors the CRF+RASP system.

It is important to note though that the performances for both systems were substantially lower than the ones achieved using the large and noisy automatically generated corpus of abstracts. This can be attributed to the fact that both systems have better performance in recognizing seen gene names rather than unseen ones. Given that the automatically generated corpus required no manual annotation and very little effort compared to the manually annotated one, it is a strong argument for bootstrapping techniques.

A known way of reducing the effect of noise in sequential models such as CRFs is to reduce their order. However, this limits the context taken into account, potentially harming the performance on unseen gene names. Keeping the same feature set, we

		HMM	CRF+RASP
overall	Recall	52.65	49.88
	Precision	46.56	72.77
	F-score	49.42	59.19
seen genes	Recall	96.49	47.37
	Precision	58.51	55.1
	F-score	72.85	50.94
unseen genes	Recall	51.4	49.95
	Precision	46.04	73.4
	F-score	48.57	59.45
seen genes		57	
unseen genes		2000	

Table 4: Results on training on manually annotated abstracts and testing on full papers

trained a first order CRF model on the noisy abstracts corpus and we evaluated on the manually annotated abstracts and full papers. As expected, the performance on the seen gene names improved but deteriorated on the unseen ones. In particular, when evaluating on abstracts the F-scores achieved were 93.22 and 38.1 respectively (compared to 84.8 and 47.02) and on full papers 86.64 and 59.86 (compared to 82.18 and 65.03). The overall performance improved substantially for the abstract where the seen genes are the majority (74.72 to 80.69), but only marginally for the more balanced full papers (72.73 to 72.89).

Ideally, we wouldn't want to sacrifice the performance on unseen genes of the CRF+RASP system in order to deal with noise. While the large noisy training dataset provides good coverage of the possible gene names, it is unlikely to contain every gene name we would encounter, as well as all the possible common English words which can become precision errors. Therefore we attempted to combine the two NER systems based on the evaluation presented earlier. Since the HMM-based system is performing very well on seen gene names, for each sentence we check whether it has recognized any gene names unseen in the training data (potential unseen precision errors) or if it considered as ordinary English words any tokens not seen as such in the training data (potential unseen recall errors). If either of these is true, then we pass the sentence to the CRF+RASP system, which has better performance on unseen gene names.

Such a strategy is expected to trade some of the performance of the seen gene names of the HMM-based system for improved performance on the unseen gene names by using the predictions of the CRF+RASP system. This occurs because in the same sentence seen and unseen gene names may coexist and choosing the predictions of the latter system could result in more errors on the seen gene names. This strategy is likely to improve the performance on datasets where there are more unseen gene names and the difference in the performance of the CRF+RASP on them is substantially better than the HMM-based. Indeed, using this strategy we achieved 73.95 overall F-score on the full paper corpus which contains slightly more unseen gene names (57% of the total gene names). For the corpus of manually annotated abstracts the performance was reduced to 80.21, which is expected since the majority of gene names (69%) are seen in the training data. and the performance of the CRF+RASP system on the unseen data is better only by a small margin than the HMM-based one (47.02 vs 44.98 in F-score respectively).

5 Discussion - Related work

The experiments of the previous section are to our knowledge the first to evaluate biomedical named entity recognition on full papers. Furthermore, we consider that using abstracts as the training material for such evaluation is a very realistic scenario, since abstracts are generally publicly available and therefore easy to share and distribute with a trainable system, while full papers on which they are usually applied are not always available.

Differences between abstracts and full papers can be important when deciding what kind of material to annotate for a certain purpose. For example, if the annotated material is going to be used as training data and given that higher coverage of gene names in the training data is beneficial, then it might be preferable to annotate abstracts because they contain greater variety of gene names which would result in higher coverage in the dataset. On the other hand, full papers contain a greater variety of contexts which can be useful for training a system and as mentioned earlier, they can be more appropriate

for evaluation.

It would be of interest to train NER systems on training material generated from full papers. Considering the effort required in manual annotation though, it would be difficult to obtain quantities of such material large enough that would provide adequate coverage of a variety of gene names. An alternative would be to generate it automatically. However, the approach employed to generate the noisy abstracts corpus used in this paper is unlikely to provide us with material of adequate quality to train a gene name recognizer. This is because more noise is going to be introduced, since full papers are likely to contain more gene names not recorded by the curators, as well as more common English words that happen to overlap with the genes mentioned in the paper.

The aim of this paper is not about deciding on which of the two models is better but about how the datasets used affect the evaluation and how to combine the strengths of the models based on the analysis performed. In this spirit, we didn't attempt any of the improvements discussed by Vlachos & Gasperin (2006) because they were based on observations on the behavior of the HMM-based system. From the analysis presented earlier, the CRF+RASP system behaves differently and therefore it's not certain that those strategies would be equally beneficial to it.

As mentioned in the introduction, there has been a lot of work on biomedical NER, either through shared tasks or independent efforts. Of particular interest is the work of Morgan et al (2004) who bootstrapped an HMM-based gene name recognizer using FlyBase resources and evaluate on abstracts. Also of interest is the system presented by Settles (2004) which used CRFs with rich feature sets and suggested that one could use features from syntactic parsing with this model given their flexibility. Direct comparisons with these works are not possible since different datasets were used.

Finaly, combining models has been a successful way of achieving good results, such as those of Florian et al. (2003) who had the top performance in the named entity recognition shared task of CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003).

6 Conclusions- Future work

In this paper we compared two different named entity recognition systems on abstracts and full paper corpora using automatically generated training data. We demonstrated how the datasets affect the evaluation and how the two systems can be combined. Also, our experiments showed that bootstrapping using automatically annotated abstracts can be efficient even when evaluating on full papers.

As future work, it would be of interest to develop an efficient way to generate data automatically from full papers which could improve the results further. An interesting approach would be to combine dictionary-based matching with an existing NER system in order to reduce the noise. Also, different ways of combining the two systems could be explored. With constrained conditional random fields (Kristjansson et al., 2004) the predictions of the HMM on seen gene names could be added as constraints to the inference performed by the CRF.

The good performance of bootstrapping gene name recognizers using automatically created training data suggests that it is a realistic alternative to fully supervised systems. The latter have benefited from a series of shared tasks that, by providing a testbed for evaluation, helped assessing and improving their performance. Given the variety of methods that are available for generating training data efficiently automatically using extant domain resources (Morgan et al., 2004) or semi-automatically (active learning approaches like Shen et al. (2004) or systems using seed rules such as Mikheev et al. (1999)), it would be of interest to have a shared task in which the participants would have access to evaluation data only and they would be invited to use such methods to develop their systems.

References

Sophia Ananiadou and John McNaught, editors. 2006. *Text Mining in Biology and Biomedicine*. Artech House, Inc.

Christian Blaschke, Lynette Hirschman, and Alexander Yeh, editors. 2004. *Proceedings of the BioCreative Workshop*, Granada, March.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceed-*

ings of the COLING/ACL 2006 Interactive Presentation Sessions.

Ann Copestake, Peter Corbett, Peter Murray-Rust, CJ Rupp, Advaith Siddharthan, Simone Teufel, and Ben Waldron. 2006. An architecture for language processing for scientific texts. In *Proceedings of the UK e-Science All Hands Meeting 2006*.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 168–171. Edmonton, Canada.

C. Gasperin, N. Karamanis, and R. Seal. 2007. Annotation of anaphoric relations in biomedical full-text articles using a domain-relevant scheme. In *Proceedings of DAARC*.

Caroline Gasperin. 2006. Semi-supervised anaphora resolution in biomedical texts. In *Proceedings of BioNLP in HLT-NAACL*, pages 96–103.

Lynette Hirschman, Marc Colosimo, Alexander Morgan, and Alexander Yeh. 2005. Overview of biocreative task 1b: normalized gene lists. *BMC Bioinformatics*.

J. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier, editors. 2004. *Proceedings of JNLPBA, Geneva*.

Martin Krallinger and Lynette Hirschman, editors. 2007. *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, Madrid, April.

Trausti Kristjansson, Aron Culotta, Paul Viola, and Andrew McCallum. 2004. Interactive information extraction with constrained conditional random fields.

J. D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, pages 282–289.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

A. Mikheev, M. Moens, and C. Grover. 1999. Named entity recognition without gazetteers.

A. A. Morgan, L. Hirschman, M. Colosimo, A. S. Yeh, and J. B. Colombe. 2004. Gene name identification and normalization using a model organism database. *J. of Biomedical Informatics*, 37(6):396–410.

L. R. Rabiner. 1990. A tutorial on hidden markov models and selected apllications in speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 267–296. Kaufmann, San Mateo, CA.

Burr Settles. 2004. Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature Sets. *Proceedings of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications*.

D. Shen, J. Zhang, J. Su, G. Zhou, and C. L. Tan. 2004. Multi-criteria-based active learning for named entity recongition. In *Proceedings of ACL 2004*, Barcelona.

Christian Siefkes. 2006. A comparison of tagging strategies for statistical information extraction. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 149–152, New York City, USA, June. Association for Computational Linguistics.

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada.

A. Vlachos and C. Gasperin. 2006. Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of BioNLP in HLT-NAACL*, pages 138–145.

A. Vlachos, C. Gasperin, I. Lewin, and T. Briscoe. 2006. Bootstrapping the recognition and anaphoric linking of named entities in drosophila articles. In *Proceedings of PSB 2006*.

Andreas Vlachos. 2007. Tackling the BioCreative2 Gene Mention task with Conditional Random Fields and Syntactic Parsing. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

Ian H. Witten and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094.

Unsupervised Learning of the Morpho-Semantic Relationship in MEDLINE®

W. John Wilbur
National Center for Biotechnology Information / National Library of Medicine, National Institutes of Health, Bethesda, MD, U.S.A.
wilbur@ncbi.nlm.nih.gov

Abstract

Morphological analysis as applied to English has generally involved the study of rules for inflections and derivations. Recent work has attempted to derive such rules from automatic analysis of corpora. Here we study similar issues, but in the context of the biological literature. We introduce a new approach which allows us to assign probabilities of the semantic relatedness of pairs of tokens that occur in text in consequence of their relatedness as character strings. Our analysis is based on over 84 million sentences that compose the MEDLINE database and over 2.3 million token types that occur in MEDLINE and enables us to identify over 36 million token type pairs which have assigned probabilities of semantic relatedness of at least 0.7 based on their similarity as strings.

1 Introduction

Morphological analysis is an important element in natural language processing. Jurafsky and Martin (2000) define morphology as the study of the way words are built up from smaller meaning bearing units, called morphemes. Robust tools for morphological analysis enable one to predict the root of a word and its syntactic class or part of speech in a sentence. A good deal of work has been done toward the automatic acquisition of rules, morphemes, and analyses of words from large corpora (Freitag, 2005; Jacquemin, 1997; Monson, 2004; Schone and Jurafsky, 2000; Wicentowski, 2004; Xu and Croft, 1998; Yarowsky and Wicentowski, 2000). While this work is important it is mostly concerned with inflectional and derivational rules that can be derived from the study of texts in a language. While our interest is related to this work, we are concerned with the multitude of tokens that appear in English texts on the subject of biology. We believe it is clear to anyone who has examined the literature on biology that there are many tokens that appear in textual material that are related to each other, but not in any standard way or by any simple rules that have general applicability even in biology. It is our goal here to achieve some understanding of when two tokens can be said to be semantically related based on their similarity as strings of characters.

Thus for us morphological relationship will be a bit more general in that we wish to infer the relatedness of two strings based on the fact that they have a certain substring of characters on which they match. But we do not require to say exactly on what part of the matching substring their semantic relationship depends. In other words we do not insist on the identification of the smaller meaning bearing units or morphemes. Key to our approach is the ability to measure the contextual similarity between two token types as well as their similarity as strings. Neither kind of measurement is unique to our application. Contextual similarity has been studied and applied in morphology (Jacquemin, 1997; Schone and Jurafsky, 2000; Xu and Croft, 1998; Yarowsky and Wicentowski, 2000) and more generally (Means and others, 2004). String

BioNLP 2007: Biological, translational, and clinical language processing, pages 201–208,
Prague, June 2007.

similarity has also received much attention (Adamson and Boreham, 1974; Alberga, 1967; Damashek, 1995; Findler and Leeuwen, 1979; Hall and Dowling, 1980; Wilbur and Kim, 2001; Willett, 1979; Zobel and Dart, 1995). However, the way we use these two measurements is, to our knowledge, new. Our approach is based on a simple postulate: If two token types are similar as strings, but they are not semantically related because of their similarity, then their contextual similarity is no greater than would be expected for two randomly chosen token types. Based on this observation we carry out an analysis which allows us to assign a probability of relatedness to pairs of token types. This proves sufficient to generate a large repository of related token type pairs among which are the expected inflectional and derivationally related pairs and much more besides.

2 Methodology

We work with a set of 2,341,917 token types which are the unique token types that occurred throughout MEDLINE in the title and abstract record fields in November of 2006. These token types do not include a set of 313 token types that represent stop words and are removed from consideration. Our analysis consists of several steps.

2.1 Measuring Contextual Similarity

In considering the context of a token in a MEDLINE record we do not consider all the text of the record. In those cases when there are multiple sentences in the record the text that does not occur in the same sentence as the token may be too distant to have any direct bearing on the interpretation of the token and will in such cases add noise to our considerations. Thus we break the whole of MEDLINE into sentences and consider the context of a token to be the additional tokens of the sentence in which it occurs. Likewise the context of a token type consists of all the additional token types that occur in all the sentences in which it occurs. We used our own software to identify sentence boundaries (unpublished), but suspect that published and freely available methods could equally be used for this purpose. This produced 84,475,092 sentences over all of MEDLINE. While there is an advantage in the specificity that comes from considering context at the sentence level, this approach also gives rise to a problem. It is not uncommon for two terms to be related semantically, but to never occur in the same sentence. This will happen, for example, if one term is a misspelling of the other or if the two terms are alternate names for the same object. Because of this we must estimate the context of each term without regard to the occurrence of the other term. Then the two estimates can be compared to compute a similarity of context. This we accomplish using formulas of probability theory applied to our setting.

Let T denote the set of 2,341,917 token types we consider and let t_1 and t_2 be two token types we wish to compare. Then we define

$$\begin{aligned} p_c(t_1) &= \sum_{i \in T} p(t_1 \mid i)\, p(i) \text{ and} \\ p_c(t_2) &= \sum_{i \in T} p(t_2 \mid i)\, p(i) \end{aligned} \quad . \qquad (1)$$

Here we refer to $p_c(t_1)$ and $p_c(t_2)$ as contextual probabilities for t_1 and t_2, respectively. The expressions on the right sides in (1) are given the standard interpretations. Thus $p(i)$ is the fraction of tokens in MEDLINE that are equal to i and $p(t_1 \mid i)$ is the fraction of sentences in MEDLINE that contain i that also contain t_1. We make a similar computation for the pair of token types

$$\begin{aligned} p_c(t_1 \wedge t_2) &= \sum_{i \in T} p(t_1 \wedge t_2 \mid i)\, p(i) \\ &= \sum_{i \in T} p(t_1 \mid i)\, p(t_2 \mid i)\, p(i) \end{aligned} \quad . \qquad (2)$$

Here we have made use of an additional assumption, that given i, t_1 and t_2 are independent in their probability of occurrence. While independence is not true, this seems to be just the right assumption for our purposes. It allows our estimate of $p_c(t_1 \wedge t_2)$ to be nonzero even though t_1 and t_2 may never occur together in a sentence. In other words it allows our estimate to reflect what context would imply if there were no rule that says the same intended word will almost never occur twice in a single sentence,

etc. Our contextual similarity is then the mutual information based on contextual probabilities

$$conSim(t_1, t_2) = \log\left(\frac{p_c(t_1 \wedge t_2)}{p_c(t_1)\,p_c(t_2)}\right) \quad (3)$$

There is one minor practical difficulty with this definition. There are many cases where $p_c(t_1 \wedge t_2)$ is zero. In any such case we define $conSim(t_1, t_2)$ to be -1000.

2.2 Measuring Lexical Similarity

Here we treat the two token types, t_1 and t_2 of the previous section, as two ASCII strings and ask how similar they are as strings. String similarity has been studied from a number of viewpoints (Adamson and Boreham, 1974; Alberga, 1967; Damashek, 1995; Findler and Leeuwen, 1979; Hall and Dowling, 1980; Wilbur and Kim, 2001; Willett, 1979; Zobel and Dart, 1995). We avoided approaches based on edit distance or other measures designed for spell checking because our problem requires the recognition of relationships more distant than simple misspellings. Our method is based on letter ngrams as features to represent any string (Adamson and Boreham, 1974; Damashek, 1995; Wilbur and Kim, 2001; Willett, 1979). If $t = "abcdefgh"$ represents a token type, then we define $F(t)$ to be the feature set associated with t and we take $F(t)$ to be composed of i) all the contiguous three character substrings "*abc*", "*bcd*", "*cde*", "*def*", "*efg*", "*fgh*"; ii) the specially marked first trigram "*abc*!"; and iii) the specially marked first letter "*a*#". This is the form of $F(t)$ for any t at least three characters long. If t consists of only two characters, say "*ab*", we take i) "*ab*"; ii) "*ab*!"; and iii) is unchanged. If t consists of only a single character "*a*", we likewise take i) "a"; ii) "a!"; and iii) is again unchanged. Here ii) and iii) are included to allow the emphasis of the beginning of strings as more important for their recognition than the remainder. We emphasize that $F(t)$ is a set of features, not a "bag-of-words", and any duplication of features is ignored. While this is a simplification, it does have the minor drawback that different strings, e.g., "*aaab*" and "*aaaaab*", can be represented by the same set of features.

Given that each string is represented by a set of features, it remains to define how we compute the similarity between two such representations. Our basic assumption here is that the probability $p(t_2 \mid t_1)$, that the semantic implications of t_1 are also represented at some level in t_2, should be represented by the fraction of the features representing t_1 that also appear in t_2. Of course there is no reason that all features should be considered of equal value. Let F denote the set of all features coming from all 2.34 million strings we are considering. We will make the assumption that there exists a set of weights $w(f)$ defined over all of $f \in F$ and representing their semantic importance. Then we have

$$p(t_2 \mid t_1) = \sum_{f \in F(t_1) \cap F(t_2)} w(f) \Big/ \sum_{f \in F(t_1)} w(f). \quad (4)$$

Based on (4) we define the lexical similarity of two token types as

$$lexSim(t_1, t_2) = (p(t_2 \mid t_1) + p(t_1 \mid t_2))/2 \quad (5)$$

In our initial application of *lexSim* we take as weights the so-called inverse document frequency weights that are commonly used in information retrieval (Sparck Jones, 1972). If $N = 2{,}341{,}917$, the number of token types, and for any feature f, n_f represents the number of token types with the feature f, the inverse document frequency weight is

$$w(f) = \log\left(\frac{N}{n_f}\right). \quad (6)$$

This weight is based on the observation that very frequent features tend not to be very important, but importance increases on the average as frequency decreases.

2.3 Estimating Semantic Relatedness

The first step is to compute the distribution of $conSim(t_1, t_2)$ over a large random sample of pairs of token types t_1 and t_2. For this purpose we computed $conSim(t_1, t_2)$ over a random

sample of 302,515 pairs. This resulted in the value -1000, 180,845 times (60% of values). The remainder of the values, based on nonzero $p_c(t_1 \wedge t_2)$ are distributed as shown in Figure 1.

Let τ denote the probability density for $conSim(t_1, t_2)$ over random pairs t_1 and t_2. Let $Sem(t_1, t_2)$ denote the predicate that asserts that t_1 and t_2 are semantically related. Then our main assumption which underlies the method is

Postulate. For any nonnegative real number r

$$Q = \{conSim(t_1, t_2) \mid lexSim(t_1, t_2) > r \wedge \neg Sem(t_1, t_2)\} \quad (7)$$

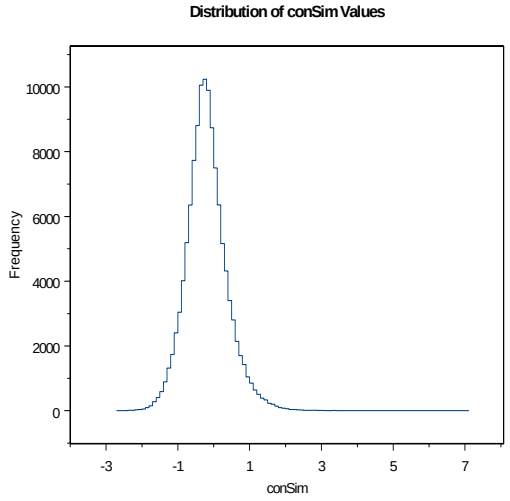


Figure 1. Distribution of *conSim* values for the 40% of randomly selected token type pairs which gave values above -1000, i.e., for which $p_c(t_1 \wedge t_2) > 0$.

has probability density function equal to τ.

This postulate says that if you have two token types that have some level of similarity as strings ($lexSim(t_1, t_2) > r$) but which are not semantically related, then $lexSim(t_1, t_2) > r$ is just an accident and it provides no information about $conSim(t_1, t_2)$.

The next step is to consider a pair of real numbers $0 \le r_1 < r_2$ and the set

$$S(r_1, r_2) = \{(t_1, t_2) \mid r_1 \le lexSim(t_1, t_2) < r_2\} \quad (8)$$

they define. We will refer to such a set as a *lexSim* slice. According to our postulate the subset of $S(r_1, r_2)$ which are pairs of tokens without a semantic relationship will produce *conSim* values obeying the τ density. We compute the *conSim* values and assume that all of those pairs that produce a *conSim* value of -1000 represent pairs that are unrelated semantically. As an example, in one of our computations we computed a slice $S(0.7, 0.725)$ and found the lexSim value -1000 produced 931,042 times. In comparing this with the random sample which produced 180,845 values of -1000, we see that

$$931{,}042 / 180{,}845 = 5.148 \quad (9)$$

So we need to multiply the frequency distribution for the random sample (shown in Figure 1) by 5.148 to represent the part of the slice $S(0.7, 0.725)$ that represents pairs not semantically related. This situation is illustrated in Figure 2. Two observations are important here. First, the two curves match almost perfectly along their left edges for *conSim* values below zero. This suggests that sematically related pairs do not produce *conSim* scores below about -1 and adds some credibility to our assumption that semantically related pairs do not produce *conSim* values of -1000. The second observation is that while the higher graph in Figure 2 represents all pairs in the *lexSim* slice and the lower graph all pairs that are not semantically related, we do not know which pairs are not semantically related. We can only estimate the probability of any pair at a particular *conSim* score level being semantically related. If we let Ψ represent the upper curve coming from the *lexSim* slice and Φ the lower curve coming from the random sample, then (10) represents the probability

$$p(x) = \frac{\Psi(x) - \Phi(x)}{\Psi(x)} \quad (10)$$

that a token type pair with a *conSim* score of x is a semantically related pair. Curve fitting or regression methods can be used to estimate p. Since it is reasonable to expect p to be a nondecreasing function of its argument, we use isotonic regression to make our estimates. For a full analysis we set

$$r_i = 0.5 + i \times 0.025 \quad (11)$$

and consider the set of *lexSim* slices $\{S(r_i, r_{i+1})\}_{i=0}^{20}$ and determine the corresponding set of probability functions $\{p_i\}_{i=0}^{20}$.

2.4 Learned Weights

Our initial step was to use the IDF weights defined in equation (6) and compute a database of all non-identical token type pairs among the 2,341,917 token types occurring in MEDLINE for which $lexSim(t_1, t_2) \geq 0.5$. We focus on the value 0.5 because the similarity measure *lexSim* has the

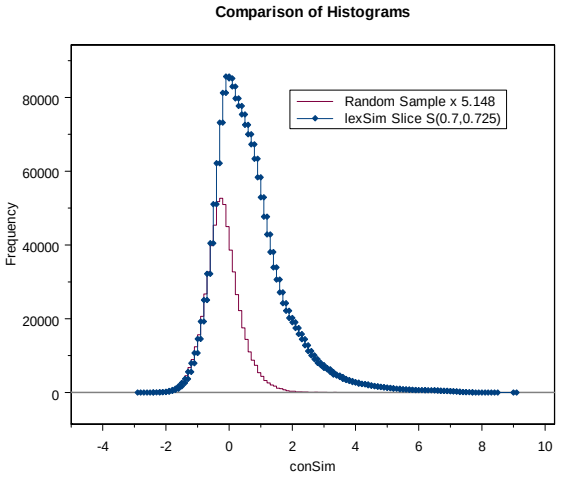


Figure 2. The distribution based on the random sample of pairs represents those pairs in the slice that are not semantically related, while the portion between the two curves represents the number of semantically related pairs.

property that if one of t_1 or t_2 is an initial segment of the other (e.g., 'glucuron' is an initial segment of 'glucuronidase') then $lexSim(t_1, t_2) \geq 0.5$ will be satisfied regardless of the set of weights used. The resulting data included the *lexSim* and the *conSim* scores and consisted of 141,164,755 pairs. We performed a complete slice analysis of this data and based on the resulting probability estimates 20,681,478 pairs among the 141,164,755 total had a probability of being semantically related which was greater than or equal to 0.7. While this seems like a very useful result, there is reason to believe the IDF weights used to compute *lexSim* are far from optimal. In an attempt to improve the weighting we divided the 141,164,755 pairs into C_{-1} consisting of 68,912,915 pairs with a *conSim* score of -1000 and C_1 consisting of the remaining 72,251,839 pairs. Letting $\vec{w}$ denote the vector of weights we defined a cost function

$$\Lambda(\vec{w}) = \sum_{(t_1,t_2)\in C_1} -\log\left(lexSim(t_1,t_2)\right) + \sum_{(t_1,t_2)\in C_{-1}} -\log\left(1 - lexSim(t_1,t_2)\right) \quad (12)$$

and carried out a minimization of Λ to obtain a set of learned weights which we will denote by $\vec{w}_0$. The minimization was done using the L-BFGS algorithm (Nash and Nocedal, 1991). Since it is important to avoid negative weights we associate a potential $v(f)$ with each ngram feature f and set

$$w(f) = \exp(v(f)). \quad (13)$$

The optimization is carried out using the potentials.

The optimization can be understood as an attempt to make *lexSim* as close to zero as possible on the large set C_{-1} where $conSim = -1000$ and we have assumed there are no semantically related pairs, while at the same time making *lexSim* large on the remainder. While this seems reasonable as a first step it is not conservative as many pairs in C_1 will not be semantically related. Because of this we would expect that there are ngrams for which we have learned weights that are not really appropriate outside of the set of 141,164,755 pairs on which we trained. If there are such, presumably the most important cases would be those where we would score pairs with inappropriately high *lexSim* scores. Our approach to correct for this possibility is to add to the initial database of 141,164,755 pairs all additional pairs which produced a $lexSim(t_1, t_2) \geq 0.5$ based on the new weight set $\vec{w}_0$. This augmented the data to a new set of 223,051,360 pairs with *conSim* scores. We then applied our learning scheme based on minimization of the function Λ to learn a new set of weights $\vec{w}_1$. There was one difference. Here and in all subsequent rounds we chose to define C_{-1} as all those pairs with

$conSim(t_1, t_2) \leq 0$ and C_1 those pairs with $conSim(t_1, t_2) > 0$. We take this to be a conservative approach as one would expect semantically related pairs to have a similar context and satisfy $conSim(t_1, t_2) > 0$ and graphs such as Figure 2 support this. In any case we view this as a conservative move and calculated to produce fewer false positives based on *lexSim* score recommendations of semantic relatedness. We actually go through repeated rounds of training and adding new pairs to the set of pairs. This process is convergent as we reach a point where the weights learned on the set of pairs does not result in the addition of a significant amount of new material. This happened with weight set $\vec{w}_4$ and a total accumulation of 440.4 million token type pairs.

Table 1. Number of token pairs and the level of their predicted probability of semantic relatedness found with three different weight sets.

Weight Set	Prob. Semantically Related ≥ 0.7	Prob. Semantically Related ≥ 0.8	Prob. Semantically Related ≥ 0.9
$\vec{w}_4$	36,173,520	22,381,318	10,805,085
Constant	34,667,988	20,282,976	8,607,863
IDF	31,617,441	18,769,424	8,516,329

3 Probability Predictions

Based on the learned weight set $\vec{w}_4$ we performed a slice analysis of the 440 million token pairs on which the weights were learned and obtained a set of 36,173,520 token pairs with predicted probabilities of being semantically related of 0.7 or greater. We performed the same slice analysis on this 440 million token pair set with the IDF weights and the set of constant weights all equal to 1. The results are given in Table 1. Here it is interesting to note that the constant weights perform substantially better than the IDF weights and come close to the performance of the $\vec{w}_4$ weights. While the $\vec{w}_4$ predicted about 1.5 million more relationships at the 0.7 probability level, it is also interesting to note that the difference between the $\vec{w}_4$ and constant weights actually increases as one goes to higher probability levels so that the learned weights allow us to predict over 2 million more relationships at the 0.9 level of reliability. This is more than a 25% increase at this high reliability level and justifies the extra effort in learning the weights.

Table 2. A table showing 30 out of a total of 379 tokens predicted to be semantically related to 'lacz' and the estimated probabilities. Ten entries are from the beginning of the list, ten from the middle, and ten from the end. Breaks where data was omitted are marked with asterisks.

Probability Semantic Relation	Token 1	Token 2
0.973028	lacz	'lacz
0.975617	lacz	010cblacz
0.963364	lacz	010cmvlacz
0.935771	lacz	07lacz
0.847727	lacz	110cmvlacz
0.851617	lacz	1716lacz
0.90737	lacz	1acz
0.9774	lacz	1hsplacz
0.762373	lacz	27lacz
0.974001	lacz	2hsplacz
***	***	***
0.95951	lacz	laczalone
0.95951	lacz	laczalpha
0.989079	lacz	laczam
0.920344	lacz	laczam15
0.903068	lacz	laczamber
0.911691	lacz	laczatttn7
0.975162	lacz	laczbg
0.953791	lacz	laczbgi
0.995333	lacz	laczbla
0.991714	lacz	laczc141
***	***	***
0.979416	lacz	ul42lacz
0.846753	lacz	veroicp6lacz
0.985656	lacz	vglacz1
0.987626	lacz	vm5lacz
0.856636	lacz	vm5neolacz
0.985475	lacz	vtkgpedeltab8rlacz
0.963028	lacz	vttdeltab8rlacz
0.993296	lacz	wlacz
0.990673	lacz	xlacz
0.946067	lacz	zflacz

Table 3. A table showing 30 out of a total of 96 tokens predicted to be semantically related to 'nociception' and the estimated probabilities. Ten entries are from the beginning of the list, ten from the middle, and ten from the end. Breaks where data was omitted are marked with asterisks.

Probability Semantic Relation	Token 1	Token 2
0.727885	nociception	actinociception
0.90132	nociception	actinociceptive
0.848615	nociception	anticociception
0.89437	nociception	anticociceptive
0.880249	nociception	antincociceptive
0.82569	nociception	antinoceiception
0.923254	nociception	antinociceptic
0.953812	nociception	antinociceptin
0.920291	nociception	antinociceptio
0.824706	nociception	antinociceptions
***	***	***
0.802133	nociception	nociceptice
0.985352	nociception	nociceptin
0.940022	nociception	nociceptin's
0.930218	nociception	nociceptine
0.944004	nociception	nociceptinerg
0.882768	nociception	nociceptinergic
0.975783	nociception	nociceptinnh2
0.921745	nociception	nociceptins
0.927747	nociception	nociceptiometric
0.976135	nociception	nociceptions
***	***	***
0.88983	nociception	subnociceptive
0.814733	nociception	thermoantinociception
0.939505	nociception	thermonociception
0.862587	nociception	thermonociceptive
0.810878	nociception	thermonociceptor
0.947374	nociception	thermonociceptors
0.81756	nociception	tyr14nociceptin
0.981115	nociception	visceronociception
0.957359	nociception	visceronociceptive
0.862587	nociception	withnociceptin

A sample of the learned relationships based on the $\vec{w}_4$ weights is contained in

Table 2 and Table 3. The symbol 'lacz' stands for a well known and much studied gene in the E. coli bacterium. Due to its many uses it has given rise to myriad strings representing different aspects of molecules, systems, or methodologies derived from or related to it. The results are not typical of the inflectional or derivational methods generally found useful in studying the morphology of English. Some might represent misspellings, but this is not readily apparent by examining them. On the other hand 'nociception' is an English word found in a dictionary and meaning "a measurable physiological event of a type usually associated with pain and agony and suffering" (Wikepedia). The data in Table 3 shows that 'nociception' is related to the expected inflectional and derivational forms, forms with affixes unique to biology, readily apparent misspellings, and foreign analogs.

4 Discussion & Conclusions

There are several possible uses for the type of data produced by our analysis. Words semantically related to a query term or terms typed by a search engine user can provide a useful query expansion in either an automatic mode or with the user selecting from a displayed list of options for query expansion. Many misspellings occur in the literature and are disambiguated in the token pairs produced by the analysis. They can be recognized as closely related low frequency-high frequency pairs. They may allow better curation of the literature on the one hand or improved spelling correction of user queries on the other. In the area of more typical language analysis, a large repository of semantically related pairs can contribute to semantic tagging of text and ultimately to better performance on the semantic aspects of parsing. Also the material we have produced can serve as a rich source of morphological information. For example, inflectional and derivational transformations applicable to the technical language of biology are well represented in the data.

There is the possibility of improving on the methods we have used, while still applying the general approach. Either a more sensitive *conSim* or *lexSim* measure or both could lead to superior results. While it is unclear to us how *conSim* might be improved, it seems there is more potential with *lexSim*. *lexSim* treats features as basically independent contributors to the similarity of token types and this is not ideal. For example the feature 'hiv' usually refers to the hu-

man immunodeficiency virus. However, if 'ive' is also a feature of the token we may well be dealing with the word 'hive' which has no relation to a human immunodeficiency virus. Thus a more complicated model of the lexical similarity of strings could result in improved recognition of semantically related strings.

In future work we hope to investigate the application of the approach we have developed to multi-token terms. We also hope to investigate the possibility of more sensitive *lexSim* measures for improved performance.

Acknowledgment This research was supported by the Intramural Research Program of the National Center for Biotechnology Information, National Library of Medicine, NIH, Bethesda, MD, USA.

References

Adamson, G. W., and Boreham, J. 1974. The use of an association measure based on character structure to identify semantically related pairs of words and document titles. Information Storage and Retrieval, 10: 253-260.

Alberga, C. N. 1967. String similarity and misspellings. Communications of the ACM, 10: 302-313.

Damashek, M. 1995. Gauging similarity with n-grams: Language-independent categorization of text. Science, 267: 843-848.

Findler, N. V., and Leeuwen, J. v. 1979. A family of similarity measures between two strings. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1: 116-119.

Freitag, D. 2005. Morphology Induction From Term Clusters, 9th Conference on Computational Natural Language Learning (CoNLL): Ann Arbor, Michigan, Association for Computational Linguistics.

Hall, P. A., and Dowling, G. R. 1980. Approximate string matching. Computing Surveys, 12: 381-402.

Jacquemin, C. 1997. Guessing morphology from terms and corpora, in Belkin, N. J., Narasimhalu, A. D., and Willett, P., editors, 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval: Philadelphia, PA, ACM Press, p. 156-165.

Jurafsky, D., and Martin, J. H. 2000. Speech and Language Processing: Upper Saddle River, New Jersey, Prentice Hall.

Means, R. W., Nemat-Nasser, S. C., Fan, A. T., and Hecht-Nielsen, R. 2004. A Powerful and General Approach to Context Exploitation in Natural Language Processing, HLT-NAACL 2004: Workshop on Computational Lexical Semantics Boston, Massachusetts, USA, Association for Computational Linguistics.

Monson, C. 2004. A framework for unsupervised natural language morphology induction, Proceedings of the ACL 2004 on Student research workshop: Barcelona, Spain, Association for Computational Linguistics.

Nash, S. G., and Nocedal, J. 1991. A numerical study of hte limited memory BFGS method and hte truncated-Newton method for large scale optimization. SIAM Journal of Optimization, 1: 358-372.

Schone, P., and Jurafsky, D. 2000. Knowledge-free induction of morphology using latent semantic analysis, Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7: Lisbon, Portugal, Association for Computational Linguistics.

Sparck Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. The Journal of Documentation, 28: 11-21.

Wicentowski, R. 2004. Multilingual Noise-Robust Supervised Morphological Analysis using the WordFrame Model, SIGPHON: Barcelona, Spain, Association for Computational Linguistics.

Wilbur, W. J., and Kim, W. 2001. Flexible phrase based query handling algorithms, in Aversa, E., and Manley, C., editors, Proceedings of the ASIST 2001 Annual Meeting: Washington, D.C., Information Today, Inc., p. 438-449.

Willett, P. 1979. Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of index terms. Journal of Documentation, 35: 296-305.

Xu, J., and Croft, W. B. 1998. Corpus-based stemming using cooccurrence of word variants. ACM TOIS, 16: 61-81.

Yarowsky, D., and Wicentowski, R. 2000. Minimally supervised morphological analysis by multimodal alignment, Proceedings of the 38th Annual Meeting on Association for Computational Linguistics: Hong Kong, Association for Computational Linguistics.

Zobel, J., and Dart, P. 1995. Finding approximate matches in large lexicons. Software-Practice and Experience, 25: 331-345.

Reranking for Biomedical Named-Entity Recognition

Kazuhiro Yoshida[*] **Jun'ichi Tsujii**[*†‡]
[*]Department of Computer Science, University of Tokyo
[†]School of Informatics, University of Manchester
[‡]National Center for Text Mining
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033 JAPAN
{kyoshida, tsujii}@is.s.u-tokyo.ac.jp

Abstract

This paper investigates improvement of automatic biomedical named-entity recognition by applying a reranking method to the COLING 2004 JNLPBA shared task of bio-entity recognition. Our system has a common reranking architecture that consists of a pipeline of two statistical classifiers which are based on log-linear models. The architecture enables the reranker to take advantage of features which are globally dependent on the label sequences, and features from the labels of other sentences than the target sentence. The experimental results show that our system achieves the labeling accuracies that are comparable to the best performance reported for the same task, thanks to the 1.55 points of F-score improvement by the reranker.

1 Introduction

Difficulty and potential application of biomedical named-entity recognition has attracted many researchers of both natural language processing and bioinformatics. The difficulty of the task largely stems from a wide variety of named entity expressions used in the domain. It is common for practical protein or gene databases to contain hundreds of thousands of items. Such a large variety of vocabulary naturally leads to long names with productive use of general words, making the task difficult to be solved by systems with naive Markov assumption of label sequences, because such systems must perform their prediction without seeing the entire string of the entities.

Importance of the treatment of long names might be implicitly indicated in the performance comparison of the participants of JNLPBA shared task (Kim et al., 2004), where the best performing system (Zhou and Su, 2004) attains their scores by extensive post-processing, which enabled the system to make use of global information of the entity labels. After the shared task, many researchers tackled the task by using conditional random fields (CRFs) (Lafferty et al., 2001), which seemed to promise improvement over locally optimized models like maximum entropy Markov models (MEMMs) (McCallum et al., 2000). However, many of the CRF systems developed after the shared task failed to reach the best performance achieved by Zhou et al. One of the reasons may be the deficiency of the dynamic programming-based systems, that the global information of sequences cannot be incorporated as features of the models. Another reason may be that the computational complexity of the models prevented the developers to invent effective features for the task. We had to wait until Tsai et al. (2006), who combine pattern-based post-processing with CRFs, for CRF-based systems to achieve the same level of performance as Zhou et al. As such, a key to further improvement of the performance of bio-entity recognition has been to employ global features, which are effective to capture the features of long names appearing in the bio domain.

In this paper, we use reranking architecture, which was successfully applied to the task of natural language parsing (Collins, 2000; Charniak and

BioNLP 2007: Biological, translational, and clinical language processing, pages 209–216,
Prague, June 2007.

Johnson, 2005), to address the problem. Reranking enables us to incorporate truly global features to the model of named entity tagging, and we aim to realize the state-of-the-art performance without depending on rule-based post-processes.

Use of global features in named-entity recognition systems is widely studied for sequence labeling including general named-entity tasks like CoNLL 2003 shared task. Such systems may be classified into two kinds, one of them uses a single classifier which is optimized incorporating non-local features, and the other consists of pipeline of more than one classifiers. The former includes Relational Markov Networks by Bunescu et al. (2004) and skip-edge CRFs by Sutton et al. (2004). A major drawback of this kind of systems may be heavy computational cost of inference both for training and running the systems, because non-local dependency forces such models to use expensive approximate inference instead of dynamic-programming-based exact inference. The latter, pipelined systems include a recent study by Krishnan et al. (2006), as well as our reranking system. Their method is a two stage model of CRFs, where the second CRF uses the global information of the output of the first CRF. Though their method is effective in capturing various non-local dependencies of named entities like consistency of labels, we may be allowed to claim that reranking is likely to be more effective in bio-entity tagging, where the treatment of long entity names is also a problem.

This paper is organized as follows. First, we briefly overview the JNLPBA shared task of bio-entity recognition and its related work. Then we explain the components of our system, one of which is an MEMM n-best tagger, and the other is a reranker based on log-linear models. Then we show the experiments to tune the performance of the system using the development set. Finally, we compare our results with the existing systems, and conclude the paper with the discussion for further improvement of the system.

2 JNLPBA shared task and related work

This section overviews the task of biomedical named entity recognition as presented in JNLPBA shared task held at COLING 2004, and the systems that were successfully applied to the task. The training data provided by the shared task consisted of 2000 abstracts of biomedical articles taken from the GENIA corpus version 3 (Ohta et al., 2002), which consists of the MEDLINE abstracts with publication years from 1990 to 1999. The articles are annotated with named-entity BIO tags as an example shown in Table 1. As usual, 'B' and 'I' tags are for beginning and internal words of named entities, and 'O' tags are for general English words that are not named entities. 'B' and 'I' tags are split into 5 sub-labels, each of which are used to represent proteins, genes, cell lines, DNAs, cell types, and RNAs. The test set of the shared task consists of 404 MEDLINE abstracts whose publication years range from 1978 to 2001. The difference of publication years between the training and test sets reflects the organizer's intention to see the entity recognizers' portability with regard to the differences of the articles' publication years.

Kim et al. (Kim et al., 2004) compare the 8 systems participated in the shared task. The systems use various classification models including CRFs, hidden Markov models (HMMs), support vector machines (SVMs), and MEMMs, with various features and external resources. Though it is impossible to observe clear correlation between the performance and classification models or resources used, an important characteristic of the best system by Zhou et al. (2004) seems to be extensive use of rule-based post processing they apply to the output of their classifier.

After the shared task, several researchers tackled the problem using the CRFs and their extensions. Okanohara et al. (2006) applied semi-CRFs (Sarawagi and Cohen, 2004), which can treat multiple words as corresponding to a single state. Friedrich et al. (2006) used CRFs with features from the external gazetteer. Current state-of-the-art for the shared-task is achieved by Tsai et al. (2006), whose improvement depends on careful design of features including the normalization of numeric expressions, and use of post-processing by automatically extracted patterns.

IL-2	gene	expression	requires	reactive	oxygen	production	by	5-lipoxygenase	.
B-DNA	I-DNA	O	O	O	O	O	O	B-protein	O

Figure 1: Example sentence from the training data.

State name	Possible next state
BOS	B-* or O
B-protein	I-protein, B-* or O
B-cell_type	I-cell_type, B-* or O
B-DNA	I-DNA, B-* or O
B-cell_line	I-cell_line, B-* or O
B-RNA	I-RNA, B-* or O
I-protein	I-protein, B-* or O
I-cell_type	I-cell_type, B-* or O
I-DNA	I-DNA, B-* or O
I-cell_line	I-cell_line, B-* or O
I-RNA	I-RNA, B-* or O
O	B-* or O

Table 1: State transition of MEMM.

3 N-best MEMM tagger

As our n-best tagger, we use a first order MEMM model (McCallum et al., 2000). Though CRFs (Lafferty et al., 2001) can be regarded as improved version of MEMMs, we have chosen MEMMs because MEMMs are usually much faster to train compared to CRFs, which enables extensive feature selection. Training a CRF tagger with features selected using an MEMM may result in yet another performance boost, but in this paper we concentrate on the MEMM as our n-best tagger, and consider CRFs as one of our future extensions.

Table 1 shows the state transition table of our MEMM model. Though existing studies suggest that changing the tag set of the original corpus, such as splitting of O tags, can contribute to the performances of named entity recognizers (Peshkin and Pfefer, 2003), our system uses the original tagset of the training data, except that the 'BOS' label is added to represent the state before the beginning of sentences.

Probability of state transition to the i-th label of a sentence is calculated by the following formula:

$$P(l_i|l_{i-1}, S) = \frac{exp(\sum_j \lambda_j f_j(l_i, l_{i-1}, S))}{\sum_l exp(\sum_j \lambda_j f_j(l, l_{i-1}, S))}. \quad (1)$$

Features used	Forward tagging	Backward tagging
unigrams, bigrams and previous labels	(62.43/71.77/66.78)	(66.02/74.73/70.10)
unigrams and bigrams	(61.64/71.73/66.30)	(65.38/74.87/69.80)
unigrams and previous labels	(62.17/71.67/66.58)	(65.59/74.77/69.88)
unigrams	(61.31/71.81/66.15)	(65.61/75.25/70.10)

Table 2: (Recall/Precision/F-score) of forward and backward tagging.

where l_i is the next BIO tag, l_{i-1} is the previous BIO tag, S is the target sentence, and f_j and l_j are feature functions and parameters of a log-linear model (Berger et al., 1996). As a first order MEMM, the probability of a label l_i is dependent on the previous label l_{i-1}, and when we calculate the normalization constant in the right hand side (i.e. the denominator of the fraction), we limit the range of l to the possible successors of the previous label. This probability is multiplied to obtain the probability of a label sequence for a sentence:

$$P(l_{1...n}|S) = \prod_i P(l_i|l_{i-1}). \quad (2)$$

The probability in Eq. 1. is estimated as a single log-linear model, regardless to the types of the target labels.

N-best tag sequences of input sentences are obtained by well-known combination of the Viterbi algorithm and A* algorithm. We implemented two methods for thresholding the best sequences: N-*best* takes the sequences whose ranks are higher than N, and θ-*best* takes the sequences that have probability higher than that of the best sequences with a factor θ, where θ is a real value between 0 and 1. The θ-best method is used in combination with N-best to limit the maximum number of selected sequences.

3.1 Backward tagging

There remains one significant choice when we develop an MEMM tagger, that is, the direction of tagging. The results of the preliminary experiment with

forward and backward MEMMs with word unigram and bigram features are shown in Table 2. (The evaluation is done using the same training and development set as used in Section 5.) As can be seen, the backward tagging outperformed forward tagging by a margin larger than 3 points, in all the cases.

One of the reasons of these striking differences may be long names which appear in biomedical texts. In order to recognize long entity names, forward tagging is preferable if we have strong clues of entities which appear around their left boundaries, and backward tagging is preferable if clues appear at right boundaries. A common example of this effect is a gene expression like 'XXX YYY gene.' The right boundary of this expression is easy to detect because of the word 'gene.' For a backward tagger, the remaining decision is only 'where to stop' the entity. But a forward tagger must decide not only 'where to start,' but also 'whether to start' the entity, before the tagger encounter the word 'gene.' In biomedical named-entity tagging, right boundaries are usually easier to detect, and it may be the reason of the superiority of the backward tagging.

We could have partially alleviated this effect by employing head-word triggers as done in Zhou et al. (2004), but we decided to use backward tagging because the results of a number of preliminary experiments, including the ones shown in Table 2 above, seemed to be showing that the backward tagging is preferable in this task setting.

3.2 Feature set

In our system, features of log-linear models are generated by concatenating (or combining) the 'atomic' features, which belong to their corresponding atomic feature classes. Feature selection is done by deciding whether to include combination of feature classes into the model. We ensure that features in the same atomic feature class do not co-occur, so that a single feature-class combination generates only one feature for each event. The following is a list of atomic feature classes implemented in our system.

Label features The target and previous labels. We also include the coarse-grained label distinction to distinguish five 'I' labels of each entity classes from the other labels, expecting smoothing effect.

Word-based features Surface strings, base forms, parts-of-speech (POSs), word shapes[1], suffixes and prefixes of words in input sentence. These features are extracted from five words around the word to be tagged, and also from the words around NP-chunk boundaries as explained bellow.

Chunk-based features Features dependent on the output of shallow parser. Word-based features of the beginning and end of noun phrases, and the distances of the target word from the beginning and end of noun phrases are used.

4 Reranker

Our reranker is based on a log-linear classifier. Given n-best tag sequences $L_i (1 \leq i \leq n)$, a log-linear model is used to estimate the probability

$$P(L_i|S) = \frac{exp(\sum_j \lambda_j f_j(L_i, S))}{\sum_k exp(\sum_j \lambda_j f_j(L_k, S))}. \qquad (3)$$

From the n-best sequences, reranker selects a sequence which maximize this probability.

The features used by the reranker are explained in the following sections. Though most of the features are binary-valued (i.e. the value of f_j in Eq. 3. is exclusively 1 or 0), the logarithm of the probability of the sequence output by the n-best tagger is also used as a real-valued feature, to ensure the reranker's improvement over the n-best tagger.

4.1 Basic features

Basic features of the reranker are straightforward extension of the features used in the MEMM tagger. The difference is that we do not have to care the locality of the features with regard to the labels.

Characteristics of words that are listed as word-based features in the previous section is also used for the reranker. Such features are chiefly extracted from around the left and right boundaries of entities. In our experiments, we used five words around the leftmost and rightmost words of the entities. We also use the entire string, affixes, word shape, concatenation of POSs, and length of entities. Some of our

[1]The shape of a word is defined as a sequence of character types contained in the word. Character types include uppercase letters, lowercase letters, numerics, space characters, and the other symbols.

features depend on two adjacent entities. Such features include the word-based features of the words between the entities, and the verbs between the entities. Most of the features are used in combination with entity types.

4.2 N-best distribution features

N-best tags of sentences other than the target sentence is available to the rerankers. This information is sometimes useful for recognizing the names in the target sentence. For example, proteins are often written as 'XXX protein' where XXX is a protein name, especially when they are first introduced in an article, and thereafter referred to simply as 'XXX.' In such cases, the first appearance is easily identified as proteins only by local features, but the subsequent ones might not, and the information of the first appearance can be effectively used to identify the other appearances.

Our system uses the distribution of the tags of the 20 neighboring sentences of the target sentence to help the tagging of the target sentence. Tag distributions are obtained by marginalizing the n-best tag sequences. Example of an effective feature is a binary-valued feature which becomes 1 when the candidate entity names in the target sentence is contained in the marginal distribution of the neighboring sentences with a probability which is above some threshold.

We also use the information of overlapping named-entity candidates which appear in the target sentence. When there is an overlap between the entities in the target sequence and any of the named-entity candidates in the marginal distribution of the target sentence, the corresponding features are used to indicate the existence of the overlapping entity and its entity type.

5 Experiments

We evaluated the performance of the system on the data set provided by the COLING 2004 JNLPBA shared-task. which consists of 2000 abstracts from the MEDLINE articles. GENIA tagger [2], a biomedical text processing tool which automatically annotates POS tags, shallow parses and named-entity tags is used to preprocess the corpus, and POS and shallow parse information is used in our experiments.

Features used	(Recall/Precision/F-score)
full set	(73.90/77.58/75.69)
w/o shallow parser	(72.63/76.35/74.44)
w/o previous labels	(72.06/75.38/73.68)

Table 3: Performance of MEMM tagger.

We divided the data into 20 contiguous and equally-sized sections, and used the first 18 sections for training, and the last 2 sections for testing while development (henceforth the training and development sets, respectively). The training data of the reranker is created by the n-best tagger, and every set of 17 sections from the training set is used to train the n-best tagger for the remaining section (The same technique is used by previous studies to avoid the n-best tagger's 'unrealistically good' performance on the training set (Collins, 2000)). Among the n-best sequences output by the MEMM tagger, the sequence with the highest F-score is used as the 'correct' sequence for training the reranker.

The two log-linear models for the MEMM tagger and reranker are estimated using a limited-memory BFGS algorithm implemented in an open-source software Amis[3]. In both models, Gaussian prior distributions are used to avoid overfitting (Chen and Rosenfeld, 1999), and the standard deviations of the Gaussian distributions are optimized to maximize the performance on the development set. We also used a thresholding technique which discards features with low frequency. This is also optimized using the development set, and the best threshold was 4 for the MEMM tagger, and 50 for the reranker [4]. For both of the MEMM tagger and reranker, combinations of feature classes are manually selected to improve the accuracies on the development set. Our final models include 49 and 148 feature class combinations for the MEMM tagger and reranker, respectively.

Table 3 shows the performance of the MEMM tagger on the development set. As reported in many

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/. The tagger is trained on the GENIA corpus, so it is likely to show very good performance on both training and development sets, but not on the test set.

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/amis/.

[4] We treated feature occurrences both in positive and negative examples as one occurrence.

Features used	(Recall/Precision/F-score)
oracle	(94.62/96.07/95.34)
full set	(75.46/78.85/77.12)
w/o features that depend on two entities	(74.67/77.99/76.29)
w/o n-best distribution features	(74.99/78.38/76.65)
baseline	(73.90/77.58/75.69)

Table 4: Performance of the reranker.

of the previous studies (Kim et al., 2004; Okanohara et al., 2006; Tzong-Han Tsai et al., 2006), features of shallow parsers had a large contribution to the performance. The information of the previous labels was also quite effective, which indicates that label unigram models (i.e. 0th order Markov models, so to speak) would have been insufficient for good performance.

Then we developed the reranker, using the results of 50-best taggers as training data. Table 4 shows the performance of the reranker pipelined with the 50-best MEMM tagger, where the 'oracle' row shows the upper bound of reranker performance. Here, we can observe that the reranker successfully improved the performance by 1.43 points from the baseline (i.e. the one-best of the MEMM tagger). It is also shown that the global features that depend on two adjacent entities, and the n-best distribution features from the outside of the target sentences, are both contributing to this performance improvement.

We also conducted experimental comparison of two thresholding methods which are described in Section 3. Since we can train and test the reranker with MEMM taggers that use different thresholding methods, we could make a table of the performance of the reranker, changing the MEMM tagger used for both training and evaluation[5].

Tables 5 and 6 show the F-scores obtained by various MEMM taggers, where the 'oracle' column again shows the performance upper bound. (All of the θ-best methods are combined with 200-best thresholding.) Though we can roughly state that the reranker can work better with n-best taggers which are more ambiguous than those used for their training, the differences are so slight to see clear tendencies (For example, the columns for the reranker trained using the 10-best MEMM tagger seems to be a counter example against the statement).

We may also be able to say that the θ-best methods are generally performing slightly better, and it could be explained by the fact that we have better oracle performance with less ambiguity in θ-best methods.

However, the scores in the column corresponding to the 50-best training seems to be as high as any of the scores of the θ-best methods, and the best score is also achieved in that column. The reason may be because our performance tuning is done exclusively using the 50-best-trained reranker. Though we could have achieved better performance by doing feature selection and hyper-parameter tuning again using θ-best MEMMs, we use the reranker trained on 50-best tags run with 70-best MEMM tagger as the best performing system in the following.

5.1 Comparison with existing systems

Table 7 shows the performance of our n-best tagger and reranker on the official test set, and the best reported results on the same task. As naturally expected, our system outperformed the systems that cannot accommodate truly global features (Note that one point of F-score improvement is valuable in this task, because inter-annotator agreement rate of human experts in bio-entity recognition is likely to be about 80%. For example, Krauthammer et al. (2004) report the inter-annotater agreement rate of 77.6% for the three way bio-entity classification task.) and the performance can be said to be at the same level as the best systems. However, in spite of our effort, our system could not outperform the best result achieved by Tsai et al. What makes Tsai et al.'s system perform better than ours might be the careful treatment of numeric expressions.

It is also notable that our MEMM tagger scored 71.10, which is comparable to the results of the systems that use CRFs. Considering the fact that the tagger's architecture is a simple first-order MEMM which is far from state-of-the-art, and it uses only POS taggers and shallow parsers as external resources, we can say that simple machine-learning-based method with carefully selected features could

[5]These results might not be a fair comparison, because the feature selection and hyper-parameter tuning are done using a reranker which is trained and tested with a 50-best tagger.

			Thresholding method for training						
Thresholding method for testing	oracle	avg. # of answers	10-best	20-best	30-best	40-best	50-best	70-best	100-best
10-best	91.00	10	76.51	76.53	76.85	76.73	77.01	76.68	76.86
20-best	93.31	20	76.40	76.55	76.83	76.62	76.95	76.68	76.85
30-best	94.40	30	76.34	76.52	76.91	76.63	77.06	76.75	76.90
40-best	94.94	40	76.39	76.58	76.91	76.71	77.14	76.75	76.92
50-best	95.34	50	76.37	76.58	76.90	76.65	77.12	76.78	76.92
70-best	95.87	60	76.38	76.57	76.91	76.71	**77.16**	76.81	76.97
100-best	96.26	70	76.38	76.59	76.95	76.74	77.10	76.82	76.98

Table 5: Comparison of the F-scores of rerankers trained and evaluated with various N-best taggers.

			Thresholding method for training						
Thresholding method for testing	oracle	avg. # of answers	0.05-best	0.02-best	0.008-best	0.004-best	0.002-best	0.0005-best	0.0002-best
0.05-best	91.65	10.7	76.70	76.80	76.93	76.64	77.02	76.78	76.52
0.02-best	93.45	17.7	76.79	76.91	77.07	76.79	77.09	76.89	76.70
0.008-best	94.81	27.7	76.79	77.01	77.05	76.80	**77.14**	76.88	76.73
0.004-best	95.55	37.5	76.79	76.98	76.97	76.74	77.12	76.86	76.71
0.002-best	96.09	49.3	76.79	76.98	76.96	76.73	77.13	76.85	76.72
0.0005-best	96.82	77.7	76.79	76.98	76.96	76.73	77.13	76.85	76.70
0.0002-best	97.04	99.2	76.83	77.01	76.96	76.71	77.13	76.88	76.70

Table 6: Comparison of the F-scores of rerankers trained and evaluated with various θ-best taggers.

	F-score	Method
This paper	71.10	MEMM
	72.65	reranking
Tsai et al. (2006)	72.98	CRF, post-processing
Zhou et al. (2004)	72.55	HMM, SVM, post-processing, gazetteer
Friedrich et al. (2006)	71.5	CRF, gazetteer
Okanohara et al. (2006)	71.48	semi-CRF

Table 7: Performance comparison on the test set.

be sufficient practical solutions for this kind of tasks.

6 Conclusion

This paper showed that the named-entity recognition, which have usually been solved by dynamic-programming-based sequence-labeling techniques with local features, can have innegligible performance improvement from reranking methods. Our system showed clear improvement over many of the machine-learning-based systems reported to date, and also proved comparable to the existing state-of-the-art systems that use rule-based post-processing.

Our future plans include further sophistication of features, such as the use of external gazetteers which is reported to improve the F-score by 1.0 and 2.7 points in (Zhou and Su, 2004) and (Friedrich et al., 2006), respectively. We expect that reranking architecture can readily accommodate dictionary-based features, because we can apply elaborated string-matching algorithms to the qualified candidate strings available at reranking phase.

We also plan to apply self-training of n-best tagger which successfully boosted the performance of one of the best existing English syntactic parser (McClosky et al., 2006). Since the test data of the shared-task consists of articles that represent the different publication years, the effects of the publication years of the texts used for self-training would be interesting to study.

References

Adam L. Berger, Stephen Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach

to Natural Language Processing. *Computational Linguistics*, 22(1).

R. Bunescu and R. Mooney. 2004. Relational markov networks for collective information extraction. In *Proceedings of ICML 2004*.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of ACL 2005*.

S. Chen and R. Rosenfeld. 1999. A Gaussian prior for smoothing maximum entropy models. In *Technical Report CMUCS*.

Michael Collins. 2000. Discriminative Reranking for Natural Language Parsing. In *Proceedings of 17th International Conference on Machine Learning*, pages 175–182. Morgan Kaufmann, San Francisco, CA.

Christoph M. Friedrich, Thomas Revillion, Martin Hofmann, and Juliane Fluck. 2006. Biomedical and Chemical Named Entity Recognition with Conditional Random Fields: The Advantage of Dictionary Features. In *Proceedings of the Second International Symposium on Semantic Mining in Biomedicine*.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the Bio-Entity Recognition Task at JNLPBA. In *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04)*, pages 70–75, Geneva, Switzerland.

Michael Krauthammer and Goran Nenadic. 2004. Term identification in the biomedical literature. *Journal of Biomedical Informatics*, 37(6).

Vijay Krishnan and Christopher D. Manning. 2006. An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. In *Proceedings of ACL 2006*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289.

Andrew McCallum, Dayne Freitag, and Fernando Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML 2000*.

D. McClosky, E. Charniak, and M. Johnson. 2006. Effective self-training for parsing. In *Proceedings of NAACL 2006*.

Tomoko Ohta, Yuka Tateisi, Hideki Mima, and Jun'ichi Tsujii. 2002. GENIA Corpus: an Annotated Research Abstract Corpus in Molecular Biology Domain. In *Proceedings of the Human Language Technology Conference (HLT 2002)*, March.

Daisuke Okanohara, Yusuke Miyao, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2006. Improving the Scalability of Semi-Markov Conditional Random Fields for Named Entity Recognition. In *Proceedings of ACL 2006*, Sydney, Australia, July.

Leonid Peshkin and Avi Pfefer. 2003. Bayesian Information Extraction Network. In *Proceedings of the Eighteenth International Joint Conf. on Artificial Intelligence*.

S. Sarawagi and W. Cohen. 2004. Semimarkov conditional random fields for information extraction. In *Proceedings of ICML 2004*.

Charles Sutton and Andrew McCallum. 2004. Collective Segmentation and Labeling of Distant Entities in Information Extraction. Technical report, University of Massachusetts. Presented at ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields.

Richard Tzong-Han Tsai, Cheng-Lung Sung, Hong-Jie Dai, Hsieh-Chuan Hung, Ting-Yi Sung, and Wen-Lian Hsu. 2006. NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. In *BMC Bioinformatics 2006, 7(Suppl 5):S11*.

GuoDong Zhou and Jian Su. 2004. Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04)*, pages 96–99.

Author Index


ACL 2007
PRAGUE